Action.NET
Guia do Usuário

Versão an-2014-1

Agosto 2014

spin ENGENHARIA DE AUTOMAÇÃO

www.spinengenharia.com.br

NOTA

Copyright 2012,2014©

Spin Engenharia de Automação Ltda

Todos os Direitos Reservados

Nenhuma parte deste documento pode ser reproduzida, copiada, fotocopiada, distribuída ou alterada sem a prévia e expressa autorização da Spin Engenharia de Automação Ltda.

Action.NET© é uma marca registrada da

Spin Engenharia de Automação Ltda.

Todas as outras marcas e nomes de produtos são marcas registradas de seus respectivos proprietários e/ou empresas.

Em diferentes partes deste documento, a empresa poderá fazer menção tanto de seu nome comercial Spin como Spin Engenharia de Automação Ltda.

Em virtude do contínuo desenvolvimento de seus produtos, a informação contida neste documento está sujeita a alterações e/ou modificações sem prévia notificação. A Spin não se considera responsável por erros de digitação ou interpretação das informações aqui contidas; e/ou por danos e prejuízos causados / gerados a terceiros. O conteúdo desta publicação poderá ser alterado a qualquer momento sem que exista a obrigação de notificar qualquer parte envolvida; isto não implicará, em nenhuma hipótese, em alterações, reclamações, ou extensão de garantia.

Conteúdo

Primeiros passos 9

Bem-vindo ao Action.NET . . . 9
Sobre este Guia . . . 10
Obtendo Ajuda . . . 10
Guia do Usuário e pesquisa no Google . . . 10
Forum On-line, E-mail e Phone . . . 11
Integradores de Sistemas e VARs . . . 11
Requisitos do Sistema . . . 11
Instalando o Action.NET . . . 12
Instalação . . . 12
Iniciando o Action.NET . . . 13
Licenciando o Action.NET . . . 14
Instalando serviços no IIS . . . 15
Procedimento de Instalação . . . 15

Visão Geral 23

A plataforma Action.NET . . . 24
O que é o Action.NET . . . 24
Vantagens Tecnológicas . . . 24
Tudo em um, completo, flexível e fácil de aplicar . . . 24
Áreas de Aplicação . . . 25
Topologia de Rede . . . 26
Arquitetura do Sistema . . . 27
Banco de dados de tempo real . . . 27
Gestão da Informação em Tempo real, TI e SCADA combinados . . . 28
Persistência e redundância . . . 29
Engenharia e Runtime . . . 29
Modelo de Objetos e Namespaces . . . 29
Projetos Action.NET . . . 31
Engenharia Distribuída . . . 32
Configuração da área de trabalho . . . 32
Módulos Action.NET . . . 33
Tags de tempo real . . . 34
Telas Gráficas . . . 35
Cinco tipos de clientes remotos são suportados . . . 35
Segurança e Redundância . . . 36
Tendência e Historiador . . . 36
Devices - Comunicação com IEDs . . . 37
Alarmes e Eventos . . . 37
Datasets e SQL . . . 37

Scripts e dot NET 38
Otimização e controle avançado 39
Relatórios e acesso a dados 39
Testando e executando projetos 40
Building 41
Testando o projeto 41
Startup do Projeto 41
Publicando o Projeto 42
Informação e Controle de Versão 43
Verificando informação sobre o Projeto 43
Controlando alterações de projeto 44
Adicionando e compartilhando anotações de projeto 44
Referência. 45
Informação de Licenciamento 45

Gerenciamento de Projetos 47

Visualizando Projetos 47
Criando um Projeto 49
Encontrando Projetos 50
Permitindo o acesso remoto 52
Accessando Projetos remotamente 53
Executando Projetos 54
Atualizando projetos 54
Configuração de projeto. 55
Ferramentas de engenharia 55
Configuração orientada para SQL 55
Multi-usuário, multi-Projeto 55
Informações sobre o Projeto 56
Configuração do Projeto (Settings). 56

Área de Trabalho de Engenharia 59

Visão geral da Área de Trabalho 59
Grid Splitter 60
Usando a barra de ferramentas 60
Desfazer e Refazer alterações 60
Cortar, Copiar e Colar inteligentes 61
Imprimir, salvar em um arquivo ou Importar 61
Encontrar objetos. 61
Nova Tag e Propriedades de uma Tag 62
Logon de segurança 62
Botão de Ajuda 62
Navegando na Área de Trabalho 62
Trabalhando com tabelas 65
Tabelas de Edição 65
Dicas e Validação 66
Classificação, filtragem e agrupamento. 66
Adicionando e removendo colunas 68
Definição de colunas comuns 68

Copiando e Colando Linhas . . . 69
Importando Dados . . . 69
Usando a Referência Cruzada . . . 70
Intellisense e alteração de Nomes . . . 71
Dicionários e Localização . . . 71
Localização da linguagem no Runtime . . . 71
Conjuntos de Enumeração . . . 73
Dicionário de Categorias . . . 74
Assistentes de Importação . . . 75
Importar projetos Action.NET . . . 75
Importar Rockwell ControlLogix . . . 75
Importar configuração de um Servidor OPC . . . 75
Importar de um sistema OSIsoft PI . . . 75
Importando Tag Templates em XML . . . 76
Importando classes dot NET de uma DLL . . . 76
Importando projetos ActionView . . . 76
Geração automatizada de projetos Action.Net . . . 76
Configurando Preferências . . . 76

Tags, Níveis e Tipos 79

Criando e Editando Tags . . . 79
Tipos de Tags pré-definidos . . . 82
Temporizador (Timer) . . . 82
Tags de Referência . . . 83
Formatos de Tags . . . 84
Criando Tipos Customizados (Tag Templates) . . . 85
Utilizando Tipos Customizados (Templates) . . . 86
Criando Níveis (Assets) . . . 88
Criando Categorias . . . 89
Objetos Tags em tempo de execução . . . 90
Tag Namespace . . . 90

Historiador e Registro de dados 91

Adicionando Tags para o Historiador . . . 91
Configurando Tabelas do Historiador . . . 92
Provedor do banco de dados Historiador . . . 93
Usando o Sistema PI da OSIsoft(tm) . . . 94
Schema de Tabelas do Historiador . . . 94
Visualizando Gráficos de Tendência . . . 95
Personalizando a Obtenção de Amostragens . . . 95
Objetos runtime do Historiador . . . 95

Segurança e Usuários 97

Adicionando e Alterando Usuários . . . 97
Configurando Permissões . . . 98
Desabilitar opção Alternar Aplicativos do Windows . . . 99
Configurando Políticas . . . 99
Removendo Usuários . . . 100

Gerenciando usuários em Tempo de execução. 101
Personalizando procedimentos de Login 101
Objetos runtime de Segurança 102

Dispositivos (Devices) e Interfaces 103

Protocolos de Comunicação 103
Protocolos Suportados 104
Interfaces adicionais 104
Toolkit para protocolos. 105
Configurando Canais 105
Configurando Nós 107
Importando de um Servidor OPC 109
Importando endereços de PLCs 109
Configurando endereços de pontos. 109
Configurando Tipos de Acessos 111
Ferramentas para diagnose. 112
Info - Informação dos módulos 112
Watch - Tags, Estados e Propriedades 113
Janela de Trace. 114
Objetos runtime Devices 115
Códigos de Status 116

Alarmes, eventos e Audit Trail 117

Configurando Audit Trail. 117
Configuração das opções de alarme. 118
Configurando grupos de alarme. 118
Configurando Itens de alarme 120
EditAlarmsItems.Condition 122
Múltiplos Itens de Alarme para um Tag. 122
Habilitar Limites por Turno 123
Visualizando Alarmes e Eventos 124
Reconhecimento de alarmes. 124
Inscrevendo-se em Notificações 124
Banco de dados de alarmes 125
Objeto Runtime Alarm. 125

Datasets e Intercâmbio de dados 127

Configurando Conexões de Bancos de dados. 127
Accessando o Microsoft Excel 130
Criando as bases de dados ODBC. 130
Métodos para se comunicar usando ODBC 131
Configurando acesso à Tabelas de BDs 133
Ler e escrever o conteúdo da tabela 134
Configurando consultas de banco de dados 134
Obtendo o conteúdo da consulta 135
Configurando Arquivos para troca de dados 135
Objetos Runtime Datasets 136

Scripts e .NET Framework 137

Configurar Tarefas . . . 137
Configurando Classes . . . 139
Trabalhando com o Editor de Código . . . 140
Configurando Expressões . . . 141
Usando o depurador dot NET . . . 143
Criando Informação de depuração . . . 143
Anexando o depurador dot NET . . . 143
Pontos de interrupção, Passo a Passo e Watch . . . 144
Objeto Runtime Script . . . 145

Telas e Símbolos 147

Seleção e Criação de Telas . . . 147
Operações em Telas . . . 148
Criando uma nova tela . . . 149
Criação de Layouts . . . 152
O exemplo de Layout StartUp . . . 154
Configurações da área de Visualização . . . 156
Clientes MS-Windows (“Desktop”) . . . 156
Cliente iOS . . . 158
O ambiente Desenhar (Draw) . . . 159
Code Behind - Scripts de Telas . . . 160
Ferramentas de desenho . . . 160
Drawing - Barra de ferramentas vertical . . . 161
Drawing - Barra de ferramentas Horizontal . . . 163
Propriedades dos Controles . . . 165
UId e Texto . . . 165
Dinâmicas e Animações . . . 166
Action Dynamic . . . 167
Dinâmica de Brilho (Shine Dynamic) . . . 169
Dinâmica Text I/O . . . 170
Dinamica de Hyperlink . . . 170
Dinâmica de Security (Segurança) . . . 171
Dinâmica Fill, Line e Text Color . . . 171
Dinâmica Bargraph . . . 172
Dinâmica Visibility . . . 173
Dinâmica MoveDrag . . . 174
Dinâmica Scale . . . 174
Dinâmica Rotate . . . 175
Dinâmica Skew . . . 175
Dinâmica TextOutput . . . 176
Controles da interface do usuário . . . 176
Configurando um painel circular . . . 177
Configurando um Gráfico de Pizza . . . 177
Configurando um Gráfico de Barras (BarChart) . . . 177
Configurando uma Janela de Alarmes . . . 178
Configurando a Janela de Tendência . . . 179
Configurando uma Janela DataGrid . . . 181
Aparência e Transformação . . . 182

Appearance 182
Transform 183
Alteração de vários controles 183
Gerenciando Imagens e Símbolos 184
Recursos de imagens 184
Gerenciando Símbolos 185
Organizando a Biblioteca de símbolos 186
Criando um Símbolo (Disjuntor) 187
Janela de Comando do Disjuntor 189
Criando um Unifilar com os Símbolos Disjuntores 190
Operações Usuais na Criação de novos Símbolos 191
Objetos runtime Display 192

Relatórios e Acesso a dados **193**
Criando Relatórios 193
Usando o Editor de Relatórios 194
Objeto runtime Report 194

Execução, Testes e Publicação de Projetos **195**
Executando a Aplicação 195
Construindo e Empacotando Projetos 196
Building projetos 197
Referências à códigos externos 198
Controle de Versão de Projetos 199
Controle das alterações de configuração 199
Publicando o Projeto 200
Objetos runtime Information 201

Implantando projetos Action.NET **203**
Sistemas locais e distribuídos 203
Instalação do produto no computador de destino 204
Licença e Verificação da configuração do Projeto 204
Instalando arquivos de configuração do Projeto 205
Configurando o servidor para Iniciar o Runtime 206
Usando um atalho na inicialização do Windows 207
Criando o atalho 208
Usando um Serviço do Windows para inicialização 209
Configuração para Clientes remotos 210
Windows Rich Clients 210
Windows Smart Clients 211
Windows Web Clients 212
Iniciando automaticamente clientes Windows 212
Clientes iOS, iPhone e iPads 213
Implantação de Sistemas Redundantes 214
Configuração das opções de redundância 214

Versões e Licenças Action.NET **217**
Números de versão do Action.NET 217

Tipos de Licenças Action.NET. 218
Família e Modelo do Produto 218
Acesso remoto à Projetos. 219
Configuração do projeto e Licenças 220
Usando várias versões Simultaneamente 220

Namespaces de tempo de execução 223

Objetos runtime Project 223
Classes do Microsoft .NET Framework. 223
Script Toolkit 223
Teclados personalizados em Telas 223
Toolkit de controles WPF. 223
API de acesso a dados COM. 223
Assistente de Importação e Toolkit de drivers. 224

Aplicação Default Enterprise 225

Objetivo 225
Objetos da Aplicação Default. 225
Layout Default 226
Telas Disponibilizadas 227
Criação dos Níveis (Assets) 228
Categorias 230
Security da Aplicação Default 230
Alarmes da Aplicação Default 231
Tabelas Historian 233

Criando aplicação a partir da Default 235

Criação dos Dicionários 235
Criação dos Templates 236
Introdução 236
Template dos Bays. 237
Criação de Níveis (Assets) 239
Criação dos tags de Bays. 240
Criação dos Devices 241
Criação de um Canal DNP3 241
Criação do Nodo. 242
Criação dos Pontos do Canal / Nodo 242
Criação dos Alarmes 243
Criação do Histórico de Medidas. 244
Criação das Telas da Aplicação 245
Tela Principal (AN_MainPage) 245
Janela de Comando (AN_Command). 249

Aplicação default Express 253

Introdução 253
Objetivo 253
Objetos da Aplicação Default Express. 254
Layout Disponível. 254

Telas Disponibilizadas 255
Tags da Aplicação Express-Default 256
Criação dos Níveis (Assets) 256
Security da Aplicação Express-Default 257
Alarms da Aplicação Express-Default 257
Historian Tables 258

Criação de Aplicação a partir Default-Express 261

Introdução 261
Criação dos Dicionários 261
Criação dos Templates 262
Criação de Níveis (Assets) 262
Criação dos tags de Bays 263
Criação do Device 264
Criação de um Canal DNP3 264
Criação do Nodo 264
Criação dos Pontos do Canal / Nodo 265
Criação dos Alarmes e Eventos 267
Tela Gráfica de Tendências e Medidas históricas 270
Criação das Telas da Aplicação 270
Janelas e Símbolos utilizados 271
Janela de Comandos (AN_Command) 271
Janela de Login (LogOn) 271
Símbolo Disjuntor 272
Simbolo Seccionadora 272
Tela Principal (AN_MainPage) 273

Utilizando ferramentas de diagnose 275

Introdução 275
Janela Watch 275
Janela Trace 276
Ativando as Ferramentas de Diagnóstico 277
Usando o Watch 278
Usando o Trace 281
Janela de Opções do Trace 281

Deploy - Geração automatizada de projetos 283

Introdução 283
Alarmes 283
Points em Devices 286
Histórico 287
Atribuição de Categorias 288
Atributos adicionais 290
Alarm Items 291
Points 292
Historian tables 292
Executando o Deploy 293

Glossário 295

Primeiros passos

As seções seguintes pretendem dar uma visão geral do Action.NET, visando familiarizar o usuário iniciante com o produto:

- "Bem-vindo ao Action.NET" na pagina 9
- "Sobre este Guia" na pagina 10
- "Obtendo Ajuda" na pagina 10
- "Instalando o Action.NET" na pagina 12
- "Licenciando o Action.NET" na pagina 14
- "Iniciando o Action.NET" na pagina 13
- "Instalando serviços no IIS" na pagina 15

Bem-vindo ao Action.NET

O Action.NET é uma poderosa ferramenta / plataforma para desenvolver e implementar sistemas de supervisão e controle de processos e equipamentos. O Action.NET oferece uma rica, única e intuitiva interface para o usuário, com gráficos dinâmicos, complementados por um editor gráfico, e vasta capacidade de aquisição de dados em tempo real, pela utilização de uma extensa biblioteca de protocolos de comunicação padrões na indústria e no setor elétrico.

O conjunto abrangente de características do Action.NET inclui também a troca de informações e de fácil acesso a dados armazenados em servidores de bancos de dados como Microsoft SQL Server, OSIsoft (tm) PI Systems, Oracle, Sybase, Informix, MySQL e outros.

O sistema inclui um amigável servidor de alarmes e módulo de notificação de eventos, módulo de registro de operações e de geração de relatórios, um avançado servidor de dados históricos, capacidades de gerenciamento de lógicas de negócios, bem como suporte para clientes locais e remotos, com base em browsers de WEB, além de suporte para utilização de dispositivos móveis como o iPad e iPhone.

Action.NET foi construído a partir do zero, sem usar qualquer código legado. É um aplicativo de código 100% "Managed" que lhe permite aproveitar e tirar proveito de todo o potencial do Microsoft. NET Framework hoje, e no futuro.

O Action.NET suporta plenamente as técnicas de "Software as a Service" (SaaS), combinadas com instalações típicas locais , permitindo-lhe acessar e colaborar em projetos de desenvolvimento em qualquer lugar, apenas com um navegador web.

Engenheiros de automação podem desenvolver projetos de SCADA / HMI (controle de supervisão e aquisição de dados / interface homem-máquina) que automatizam grande variedade de funções em uma indústria qualquer e em particular para plantas do setor elétrico.

O pessoal de TI pode desenvolver projetos que centralizam informações para enviar dados à portais para gerenciamento e análise. Além disso, um único projeto pode realizar ambas as funções de automação e de gestão.

Sobre este Guia

Este guia foi desenvolvido para usuários de Action.NET e descreve como instalar e usar o software. Supõe-se que você esteja familiarizado com sistemas de monitoração e controle e suas necessidades, bem como os objetivos gerais de softwares SCADA.. Você deve estar familiarizado com os seus sistemas e necessidades de controle.

Este guia utiliza os seguintes símbolos e convenções tipográficas:

Convenção	Descrição
Negrito	Palavras em **negrito** mostram itens para selecionar ou clicar, tais como itens de menus ou botões.
Aviso	Aviso contém informação critica que pode afetar seus dados ou sistemas
Dica	Dicas contém informação útil para melhorar o desenho e a produtividade de suas aplicações.
Nota	Notas contém informação importante que pode afetar como instalar ou configurar o seu sistema.

Obtendo Ajuda

Guia do Usuário e pesquisa no Google

O Guia do Usuário Action.NET está disponível em PDF e CHM no computador local e on-line utilizando páginas HTML padrão ou através de uma interface única-WEB integrada com o Google Translator para suporte multi-linguagem.

O ponto de entrada para todos os quatro formatos de Ajuda está localizado na área de **Info>Ajuda** na configuração do espaço de trabalho do projeto, e também é acessível clicando no ícone Ajuda ou pressionando F1.

Todos os formatos de ajudas suportam pesquisa de contexto. Você também pode fazer uma pesquisa no Google sobre o site da Spin. A vantagem da pesquisa do Google é que inclui todos os arquivos On line de ajuda, além de todo o conteúdo do site da Spin, incluindo os foruns web.

Na página do Info-Ajuda, no espaço de trabalho de Configuração do Projeto, há uma interface de usuário para permitir que você faça pesquisas Google sobre site da SPIN sem sair da janela Action.NET.

Forum On-line, E-mail e Phone

Os usuários registrados têm acesso gratuito ao Forum on-line no site de SPIN. Se você tiver alguma dificuldade de criar sua conta, envie um e-mail para support@spin.com, esta conta de e-mail também é o ponto de entrada para abrir chamadas de suporte técnico para usuários licenciados.

Se você ainda não é um usuário licenciado e está apenas avaliando o produto, você também têm direito ao suporte, envie suas perguntas e pedidos de info@spin.com.

Os usuários licenciados também têm acesso a suporte por telefone. Contate-nos para acordos especiais em relação ao suporte fora do horário comercial, suporte 24 horas, ou auxilio on-line.

Integradores de Sistemas e VARs

A SPIN tem um programa para Integradores de Sistemas e VARs para suportar totalmente as empresas que desenvolvem aplicações e soluções em cima da plataforma Action.NET. Os integradores de sistemas e VARs são a principal fonte para responder as perguntas, de usuários finais, que são específicas para as configurações e soluções criadas por eles..

Requisitos do Sistema

São os seguintes os requisitos técnicos para o Action.NET:

- Microsoft dotNET Framework v4.0
- Sistemas operacionais: qualquer sistema operacional capaz de executar o Microsoft NET Framework 4.0 ou máquinas virtuais compatíveis com o Microsoft NET Framework.
  - Windows 7, Windows 8, Windows Vista, Windows Server 2008 e Windows Server 2012 todos vêm com a Microsoft. Instalado Net.
  - Para o Windows XP, você deve instalar o Microsoft. NET Framework v4.0.
- Requisitos mínimos de Memória RAM: Runtime 1 GB; Engenharia 2 GB
- Espaço em disco-150 MB.
- Resolução da tela:
  - Para o desenvolvimento de aplicações de mínima de 1024 x 768.
  - Para a aplicação de tempo real: As Telas são independentes da resolução, para que você possa criar aplicações que vão desde pequenos HMI de 6” até grandes monitores de alta definição.
- Para ter acesso às instalações baseadas na web, Internet Explorer v8 ou posterior.

Instalando o Action.NET

O Action.NET está disponível como um download a partir do web site ou a partir de dispositivos USB. CDs também estão disponíveis, mediante solicitação. Se você baixou a versão Express apenas para fins de avaliação, não é necessária uma licença para a instalação do produto.

A versão Express permite até 75 pontos de comunicação e a execução do aplicativo em tempo real por cerca de até dua horas, para cada execução. Esta versão destina-se apenas à avaliação e aprendizagem inicial do produto;

As versões Express não estão autorizados a ser vendidas ou aplicadas em projetos para instalação em produção.

A versão completa licenciada não tem limitações de tempo de execução. Para obter informações sobre a compra de uma licença, entre em contato com seu distribuidor.

Se você tiver o Microsoft Internet Information Services (IIS) instalado no seu computador, você verá uma mensagem durante a instalação que o servidor da Web incorporado não pôde ser instalado. Todos os recursos do Action.NET podem trabalhar sem o servidor da Web incorporado, exceto para acesso remoto a projetos e servir páginas web em tempo de execução. Se você quiser usar esses recursos, você pode configurar o IIS para trabalhar com o Action.NET como descrito na seção "Instalando serviços no IIS" na pagina 15.

Use o Windows Update para se certificar de que você tem todas as últimas atualizações críticas instaladas.

Alguns anti-vírus, como o Bit-defensor, deve ser desativado antes de executar a instalação.

Instalação

1. Se você estiver usando o Windows XP, instale o Microsoft. NET Framework 4.0.
    - Se você tiver o Microsoft Office instalado, este é tipicamente já instalado. Se necessário, você pode instalá-lo a partir de http://www.microsoft.com/en-us/download/details.aspx?id=1785 .
2. Clique duas vezes no arquivo de instalação.
    - Caso algum pré-requisito de software estiver faltando, este será instalado:
        - Redistribuíveis do Microsft VC ++
        - Core components do OPC server;
        - Microsoft .NET Framework 4.0;
    - A janela de Boas-Vindas é exibida
3. Clique **em Avançar** .
    - É mostrada janela com o Contrato de Licença
5. Leia o contrato de licença e, se concordar com ele, clique em **Aceitar** e após pressione **Avançar**.
    - É mostrada a janela Os Selecionar pasta de instalação.

6. Se você quiser instalar em um diretório diferente do padrão, clique em **Browse** e navegue até onde você deseja instalar.

7. Clique **em Avançar** .

8. Clique **Instalar** .

• Quando a instalação estiver concluída, clique em **Finish** .

• Se você comprou uma licença, você pode licenciar sua instalação. Para mais informações, consulte "Licenciando o Action.NET" na pagina 14. Se você não comprar uma licença, você pode usar a versão Express para fins de avaliação e de aprendizagem.

Iniciando o Action.NET

Licenciando o Action.NET

Se você comprou uma licença, você pode licenciar sua instalação. Depois de comprar a licença, você deve registrar a chave de licença. Enquanto você não comprar uma licença, você pode usar a versão Express para treinamento e avaliação.

Para o licenciamento pode-se usar uma "Soft Key" (chave de proteção por software) ou uma hardkey (dongle). Você pode conectar uma hardkey em um computador que já tem uma Chave de software instalada. Enquanto o hardkey estiver conectado, o sistema usa as configurações de licença da hardkey.

Use o mesmo procedimento abaixo para atualizar uma licença tanto para Soft keys como para Hard keys.

Para mais informações sobre licenciamento, consulte "Versões e Licenças Action.NET" na pagina 217.

Para licenciar o Action.NET:

1. Inicie o Action.NET.
   - Veja "Iniciando o Action.NET" na pagina 13 .
2. Na página Action.NET principal, clique em **Licence** .
   - Ou ir para **Info> Licence> Register** . durante a edição de um projeto.
3. Se você estiver usando uma Hard Key, conecte-a ao seu computador.
4. Clique **Get Site Code** (Obter o código do site).
5. Navegue para onde você quer salvar o arquivo, aceite o nome de arquivo padrão e clique em **Salvar** .
6. Quando a mensagem de confirmação for exibida, clique em **OK** .
   - Envie o arquivo para o seu distribuidor ou para support@spin.com.br .
7. Depois de receber o arquivo com a chave de licença por e-mail, coloque o arquivo de chave de licença em uma pasta no seu computador ou rede.
8. Se você estiver usando uma HardKey , conecte-a ao seu computador.
9. Inicie o **Action.NET**.
10. Na página Action.NET principal, clique em **Licence** .
11. Clique **LoadLicenseKey** .
12. Navegue até a pasta com o arquivo de chave de licença e selecione o arquivo.
13. Clique **Open**
14. Verifique em Informações sobre a licença, o número de série e as novas configurações de licença.

Instalando serviços no IIS

Ao instalar o Action.NET, se o IIS foi instalado anteriormente, você não será capaz de instalar o Servidor Web embutido no produto Action.NET (TWebServer.exe).

O software Action.NET será executado sem o servidor web, a não ser para acessar remotamente configurações de projeto e servir páginas web em tempo de execução; Todas as outras ferramentas de engenharia e de execução em tempo real do Action.NET não tem necessidade de que esteja instalado o TWebServer ou de quaisquer configurações do MS- IIS.

Se você deseja habilitar o acesso remoto de engenharia e os clientes web usando o IIS, você precisa instalar alguns serviços Action.NET no IIS ou ativar o servidor web embutido TWebServer.

Outro cenário em que o IIS é necessário é quando você quer executar o módulos de tempo real como um serviço do Windows, bem como para fornecer acesso remoto a páginas web do projeto, monitores do iPad e módulo de engenharia.

A fim de permitir a execução do TWebServer você deve parar o servidor IIS ou outras aplicações que potencialmente usam a porta 80 e executar a instalação do produto.

Para configurar o servidor IIS, siga o procedimento neste capítulo.

Esta seção é a explicação de como instalar os serviços no IIS. A explicação considera a versão7.x e 8, do IIS, mas outras versões devem ser semelhantes.

Você vai instalar três serviços dentro do IIS:

- **TProjectServer**-Permite acessar remotamente suas configurações de projeto.
- **TVisualizerWeb / TVisualizerRemote**-Permite acesso remoto a telas da execução em tempo real.
- **iDataPanel** -É o serviço para fornecer dados para dispositivos iOS..

Procedimento de Instalação

1. Verifique se o IIS está instalado, vá em **Painel de Controle> Ferramentas Administrativas** e procure o Internet Information Services (IIS) Manager na lista.
2. Habilite o ASP.Net. Svc Handlers para o serviço WCF.

Na Versão 7.x do IIS:

Se o IIS foi instalado após a instalação do NET Framework. É necessário realizar um procedimento conforme descrito no http://msdn.microsoft.com/en-us/library/ms752252.aspx, que é o de executar os seguintes programas usando o prompt de comando como administrador:

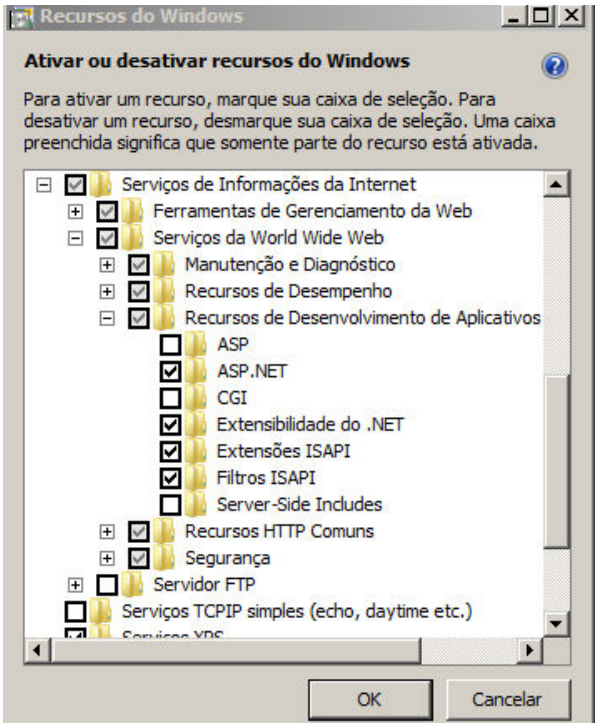
- A partir da pasta Net Microsoft, abra o Prompt de Comando e digite os seguintes comandos:.
  - aspnet_regiis.exe-i-enable
  - ServiceModelReg.exe-r

- Microsoft Net está instalado em uma das seguintes pastas:
  - C: \ Windows \ Microsoft.NET \ Framework \ v4.0.30319 (para sistemas operacionais de 32 bits)
  - C: \ Windows \ Microsoft.NET \ Framework64 \ v4.0.30319 (para sistemas operacionais de 64 bits)

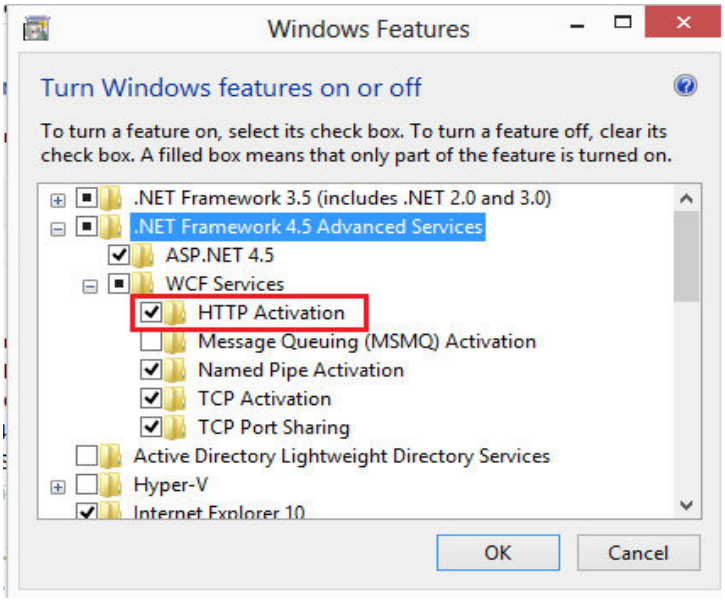
Para o IIS 8.0:

- A partir do Painel de controle, vá para **Programas e Recursos**, clique em **" Ativar/ Desativar recursos do Windows "**

Habilite a opção ASP.NET 4.5. Navegue pela árvore: **Serviços de Informações da Internet> Serviços de World Wide Web > Recursos de Desenvolvimento de Aplicativos> ASP.NET 4.5** :



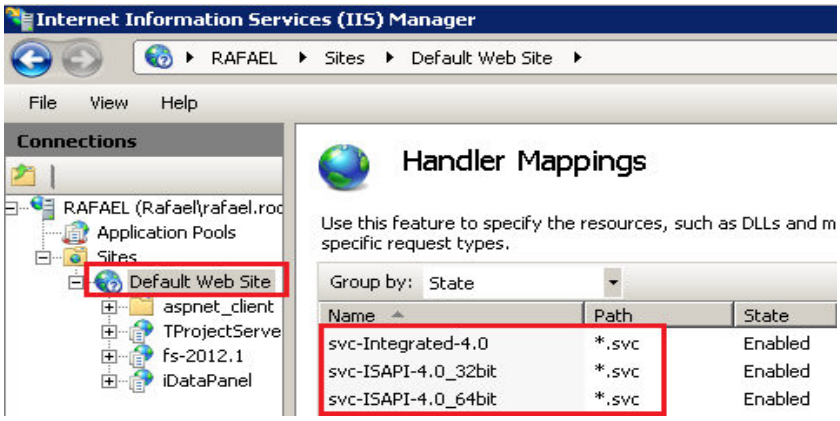
- Habilite a opção HTTP Activation (sob .Net Framework 4.5 Advanced Services).



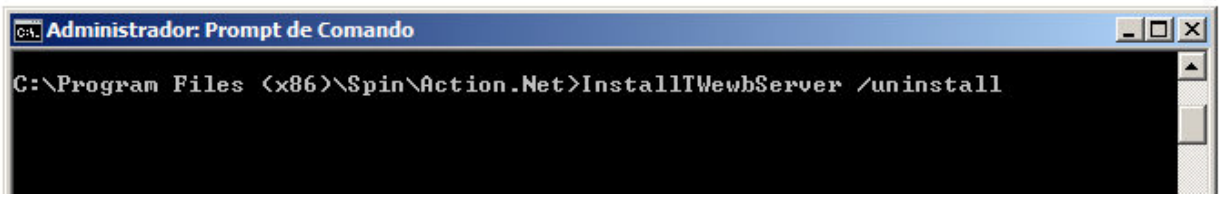
3. No IIS Manager, verifique que a extensão .svc está mapeada para o aplicativo **aspnet_isapi.dll.**

   - Se necessário siga os procedimentos descritos em http://msdn.microsoft.com/en-us/library/ms752252.aspx para a vesão correta do IIS que voce está de fato usando.

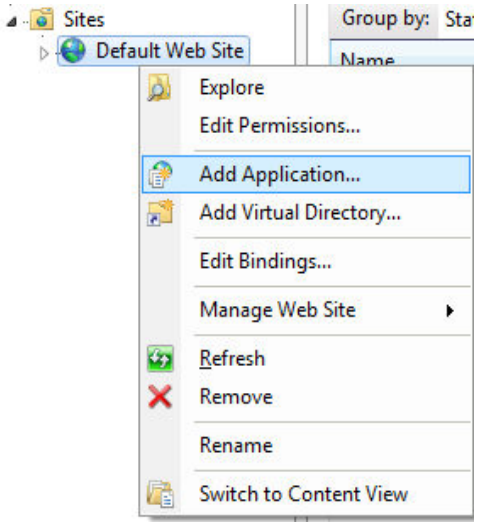
     - O diretório de instalação do.NET Framework installation é usualmente: C:\Windows\Microsoft.NET\Framework\v4.0.30319, or C:\Windows\Microsoft.NET\Framework64\v4.0.30319 (for x64 Windows)



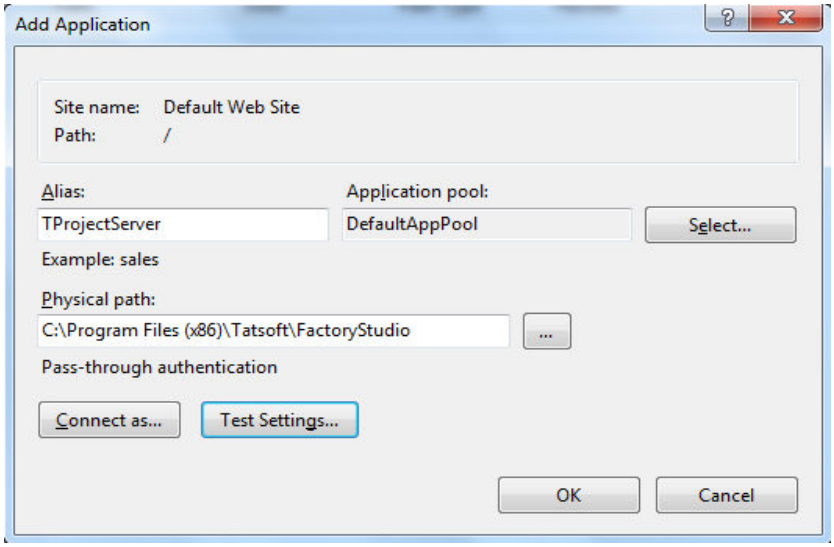
4. A partir do diretório de instalação do Action.NET, execute o utilitário: **InstallTWebServer.exe /uninstall**, que removerá a instalação default do servidor TWebServer .



5. Entre no IIS Manager, e vá para **Sites > Default Web Site > Add Application** e configure a informação para os serviços como descrito nos passos seguintes:

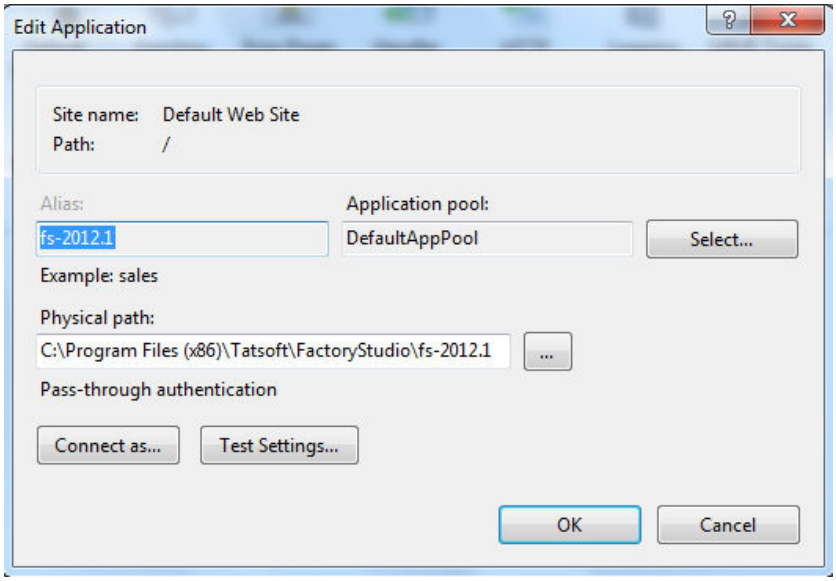


6. Para o TProjectServer, configure:
   - Alias—TProjectServer
   - Physical Path—C:\Program Files\Spin\Action.NET (Default folder. Tenha certeza de usar o caminho correto para a sua instalação do Action.NET.)
   - Application Pool—Configure um "pool" baseado no .NET 4.0. Exemplo: ASP.NET v4.0



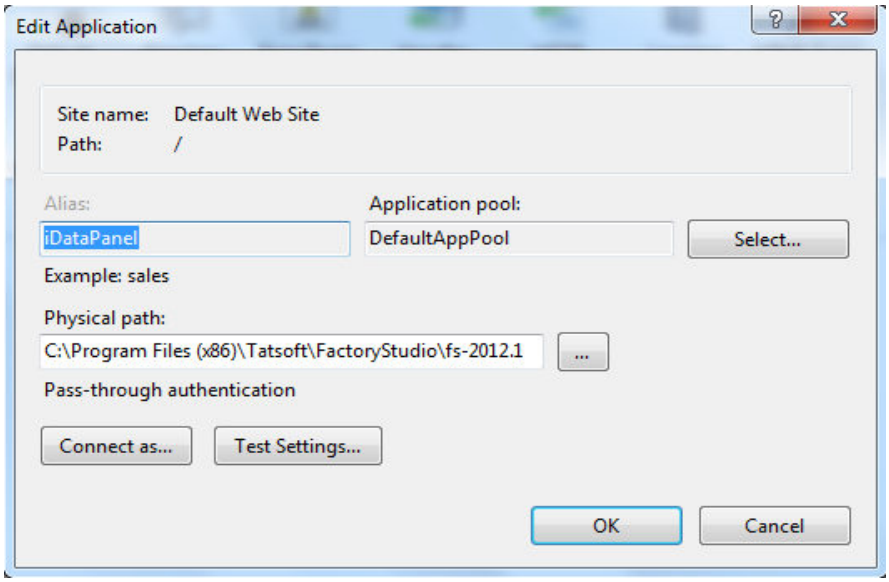
   - Nota: Security deve ser habilitado para "any user" (Everyone), pelo menos para Leitura
   - Entre no IE, e use o URL http://localhost/tprojectserver/service.svc para verificar se o serviço foi instalado corretamente. Deve ser mostrada uma página com informação sobre o serviço.
7. Para o TVisualizerWeb/TVisualizerRemote, configure:
   - Alias—an-2014.1
   - Physical Path—C:\Program Files\Spin\Action.NET\an-2014.1 (Pasta padrão Certifique-se de usar o caminho correto para a pasta de instalação do Action.NET.)

- Application Pool—Configure um "pool" baseado no .NET 4.0. Exemplo: ASP.NET

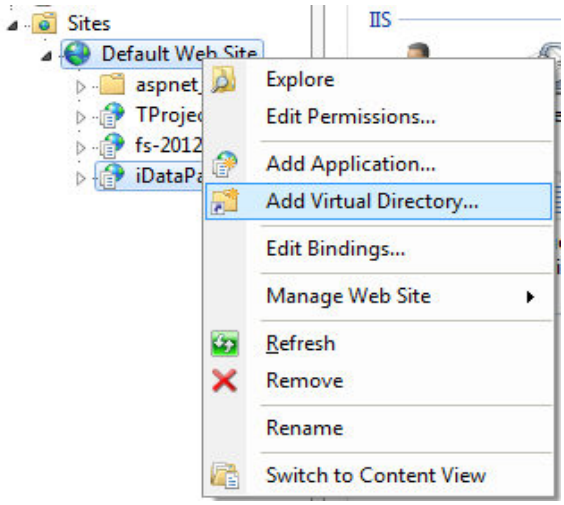


- Nota: A segurança deve permitir a qualquer utilizador (Everyone), pelo menos para a leitura.
- No IE, teste a URL http://localhost/an-2012.1/service.svc para verificar se o serviço foi instalado corretamente. Ele deve mostrar uma página com informações sobre o serviço.

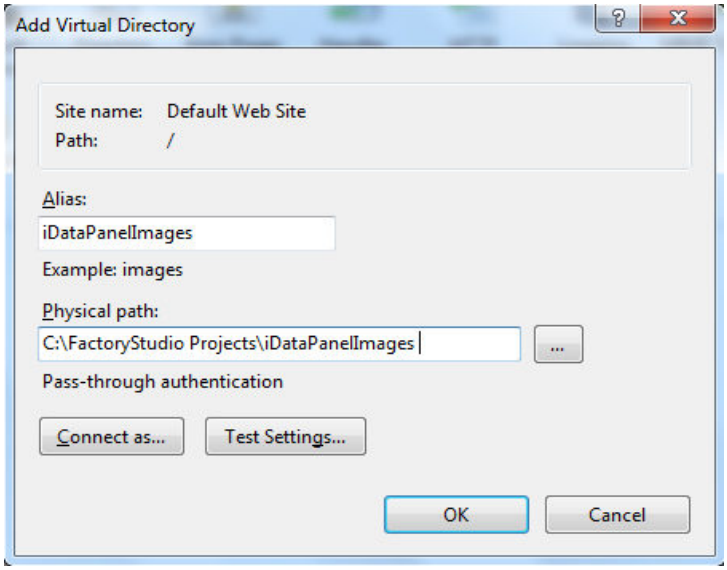
8. Para iDataPanel Service, configure:
   - Alias—iDataPanel
   - Physical Path—C:\Program Files\Spin\Action.NET\an-2014.1 (Pasta padrão Certifique-se de usar o caminho correto para a pasta de instalação do Action.NET.)
   - Application Pool—Configure um "pool" baseado no .NET 4.0. Exemplo: ASP.NET
   - Nota: Security deve ser habilitado para "any user" (Everyone), pelo menos para Leitura.



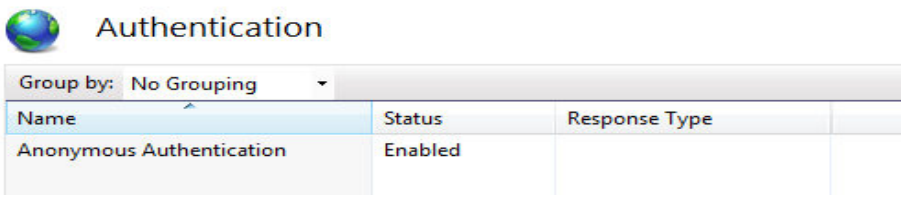
9. DataPanelImages (Virtual Directory), configure:



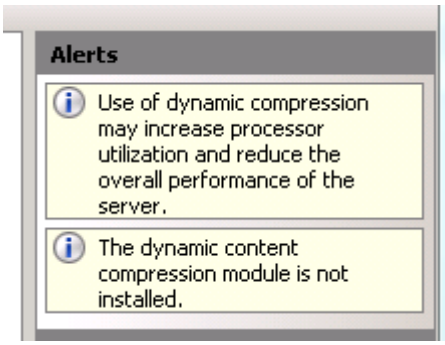
- Alias—iDataPanelImages
- Localização fisica:—C:\Action.NET Projects\iDataPanelImages



10. Habilite a Anonymous Authentication para cada um dos serviços.

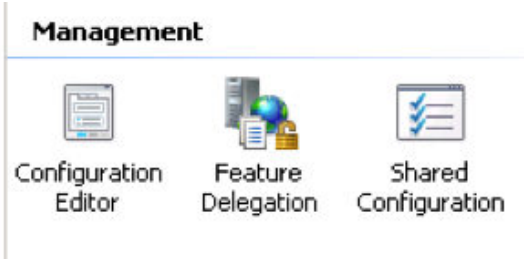


11. Configure o módulo de Compressão. Se ele não estiver instalado, você receberá textos de alerta.

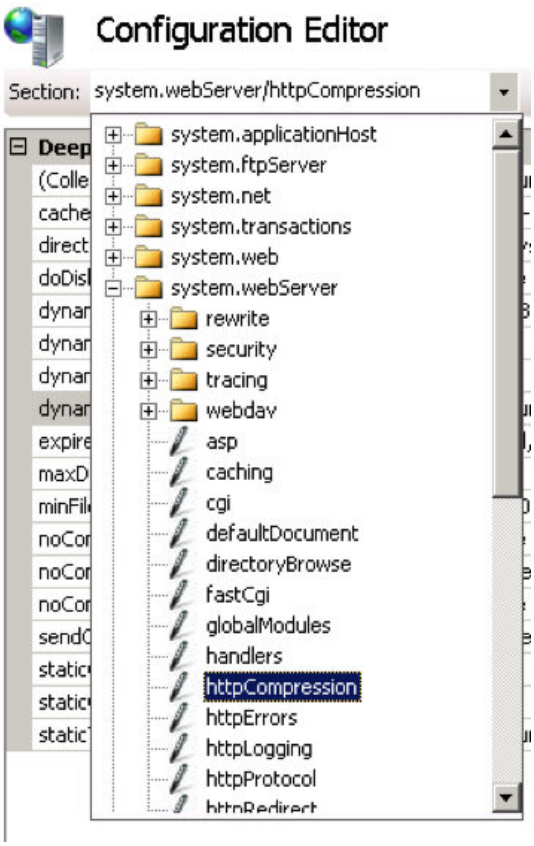


- Se não estiver instalado, vá para o gerenciador de servidores, Roles, Web Server. Em Serviços de Função, verifique as funções instaladas. Se Compressão Dinâmica não estiver instalado, clique em Adicionar Funções e instale-o

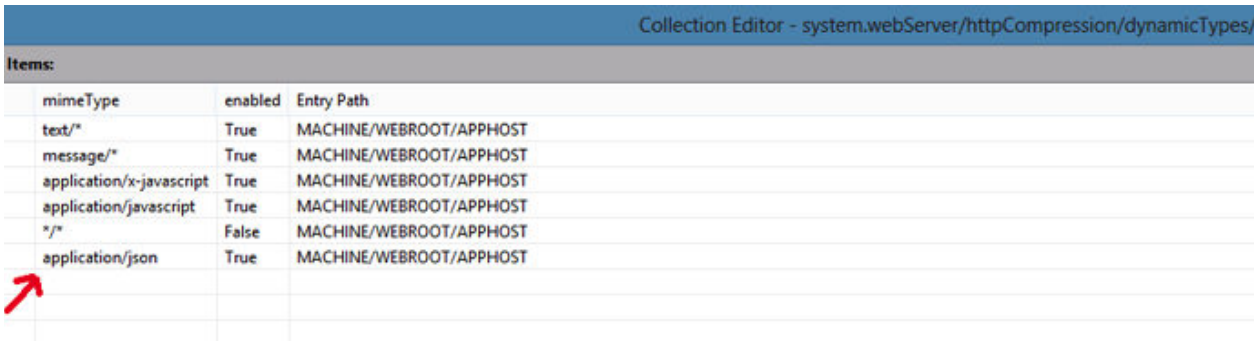
- Você pode voltar à compressão para o seu site e garantir que a Compressão Dinâmica está marcada. Neste ponto, Compressão Dinâmica deve ser configurada, mas você realmente precisa ser mais específico sobre quais mimeTypes serão comprimidos
- De volta ao IIS Manager, vá para a pagina do SERVER (not the SITE) e clique no Editor de Configuração:



- Na lista selecione **system.webServer/ webdev / httpCompression**:



- Então clique em Dynamic Types e agora que você está no Editor da Lista, adicione a aplicação/json como mostrado abaixo:



Depois de ter adicionado a aplicação e fechado a janela:

12. Se os usuários remotos não podem acessar o Action.NET, você deve permitir a todos o acesso à pasta Action.NET, pelo menos para a leitura. Clique com o botão direito do mouse na pasta, na aba **Security**, clique em **Edit**, em seguida, adicione permissões de leitura para todos.

AVISO - O IIS deve ser reiniciado após a configuração dos serviços.

Visão Geral

Este capítulo fornece uma visão geral técnica do produto Action.NET e um guia rápido para entender as áreas de aplicação e ferramentas de configuração disponíveis. O conteúdo está organizado nas seguintes seções:

- "A plataforma Action.NET" na pagina 24 contém informações sobre:
  - "O que é o Action.NET" na pagina 24,
  - "Áreas de Aplicação" na pagina 25,
  - "Topologia de Rede" na pagina 26
- "Arquitetura do Sistema" na pagina 27 contém informação sobre arquitetura;
- "Banco de dados de tempo real" na pagina 27
  - "Modelo de Objetos e Namespaces" na pagina 29
- "Projetos Action.NET" na pagina 31 apresenta a Seleção de Projetos e a configuração da interface do usuário.
- "Módulos Action.NET" na pagina 33 contém informações sobre os principais módulos, incluindo:
  - "Tags de tempo real" na pagina 34
  - "Telas Gráficas" na pagina 35
  - "Segurança e Redundância" na pagina 36
  - "Tendência e Historiador" na pagina 36
  - "Devices - Comunicação com IEDs" na pagina 37
  - "Alarmes e Eventos" na pagina 37
  - "Datasets e SQL" na pagina 37
  - "Scripts e dot NET" na pagina 38
  - "Relatórios e acesso a dados" na pagina 39
- "Testando e executando projetos" na pagina 40 expõe os conceitos básicos sobre a execução e testes de uma aplicação
- "Informação e Controle de Versão" na pagina 43 explica sobre algumas principais facilidades para o gerenciamento do ciclo de vida de um projeto.

A plataforma Action.NET

O que é o Action.NET

O Action.NET é uma poderosa plataforma para desenvolver e implementar aplicativos para o gerenciamento de informações em tempo real, fornecendo um conjunto completo de módulos em uma interface de usuário de engenharia unificada e intuitiva. Ele inclui um banco de dados com os tags em tempo real, os níveis hierárquicos de ativos e modelos, alarmes e eventos, historiador, receitas, consultas SQL e acesso de dados, elaboração de relatórios, lógicas em scripts em linguagem .NET, cliente e servidor OPC, WCF e protocolos nativos da indústria, gráficos dinâmicos criados em WPF e acessível a partir de desktops, clientes remotos inteligentes, navegador e iOS clientes nativos em iPads e iPhones.

Action.NET é um software SCADA, com uma infra-estrutura flexível para gerenciamento de dados em tempo real, tem sido aplicado ao Setor Elétrico, em geração, transmissão e distribuição, Energia Renovável, e outras plantas distribuídas como o gerenciamento de distribuição de água e sistemas de Automação Predial.

Vantagens Tecnológicas

A arquitetura do Action.NET foi criada por uma equipe de profissionais com mais de 20 anos de experiência no desenvolvimento de softwares a partir do zero e inteiramente sem o emprego de qualquer código legado; Isto que tornou possível a criação de um sistema que é simples de usar, mas muito poderoso e flexível. É um aplicativo de código inteiramente Microsoft dot NET. Que capacita os usuários finais e engenheiros de sistemas alavancar e tirar o máximo partido das tecnologias mais atuais.

Tudo em um, completo, flexível e fácil de aplicar

Um dos conceitos fundamentais Action.NET baseia-se, no fornecimento de uma solução completa e unificada, para que os usuários não tenham que construir quebra-cabeças antes de iniciar seus projetos.

O projeto unificado a partir do zero permite que você crie um sistema muito flexível, mas mantendo a facilidade de uso e design intuitivo.

Áreas de Aplicação

Controle de Processos: sistemas de controle necessitam de componentes servidores confiáveis e a Tela de telas dedicadas e bem elaboradas, para que se possa maximizar a segurança da operação, e obter o máximo do poder dos equipamentos de computação. O software Action.NET tem um conjunto completo de funcionalidades para implementar aplicações de controle de processos distribuídas e de missão critica.

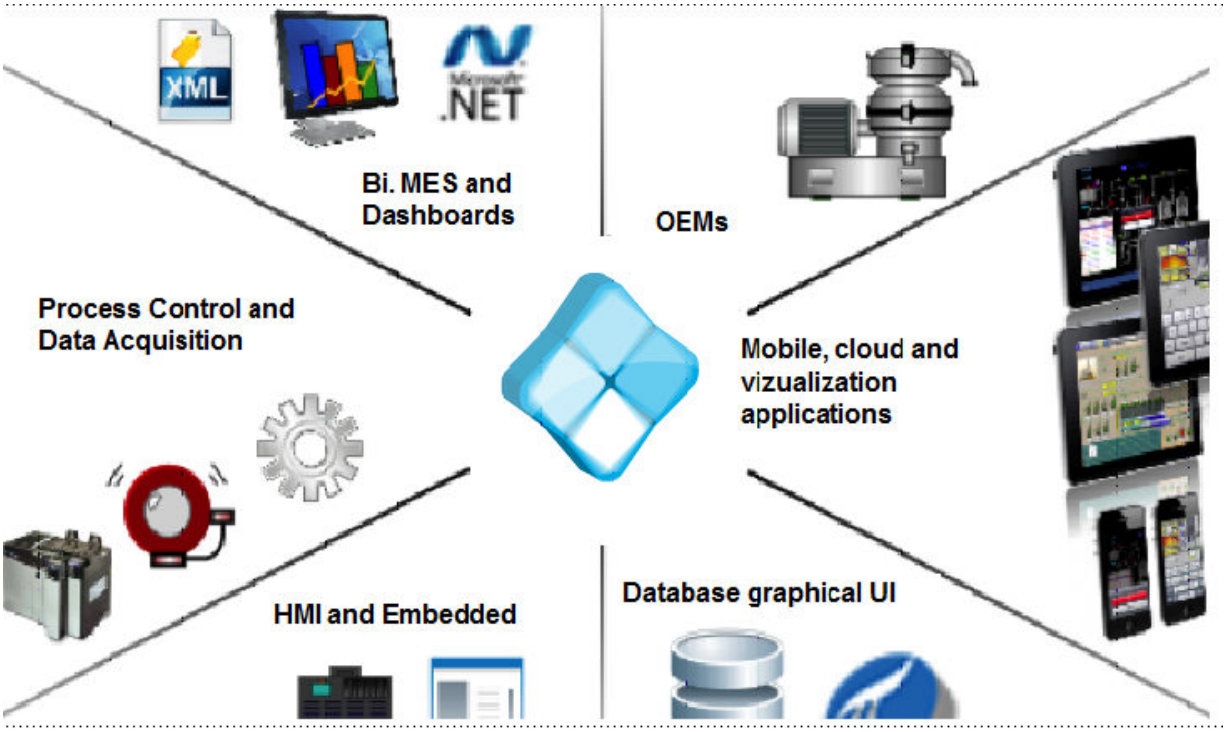
As aplicações nativas para iPads e iPhones : um aplicativo nativo pode ter funcionalidade e qualidade de uma interface de usuário que uma página web simplesmente não pode igualar. O Action.NET permite que as mesmas telas criadas para desktops também sejam disponíveis para serem executadas nativamente em dispositivos iOS, incluindo a opção da AppleStore.

SQL e OSIsoft (tm) PI System front-end: O Action.NET tem muitos recursos para gerenciar bancos de dados SQL, tabelas e consultas em tempo real. Ele também tem uma sincronização de dados com nível SDK OSIsoft (tm) Sistema PI e PI AF (tm) . Isso faz com que ele seja a melhor plataforma SCADA quando você precisa para criar gráficos personalizados dinâmicos e front-ends para esses sistemas.

HMI, embutidos e OEMs : Action.NET tem uma vasta gama de protocolos PLC built-in e uma versão especial para visualização avançada em HMI e sistemas embarcados. O sistema de controle de versão, pouco requisitos de espaço, e facilidade de acesso remoto fazem do Action.NET a melhor opção para construtores de máquinas e OEMs.

Monitoramento do Desempenho: A capacidade de criar uma camada de apresentação que é fonte independente e "ready-to-cloud", faz do Action.NET uma excelente ferramenta para criar painéis operacionais, historiadores, Manufacturing Intelligence (MI), painéis operacionais, OEE e cálculos KPI e aplicações conhecedoras do ambiente.

SCADA, MES e PIMS: Além de ter todo o espectro de funcionalidades esperadas para sistemas SCADA, O Action.NET estendeu os conceitos de banco de dados em tempo real para incluir objetos avançados, tais como tabelas de dados, matrizes dinâmicas e imagens, eventos de tempo, todos com uma forte conexão à infraestrutura do dot NET. Isto faz que seja a melhor plataforma para Sistemas de Gestão de Informação de Plantas ou para qualquer aplicação visando a gestão e apresentação de informações em tempo real.

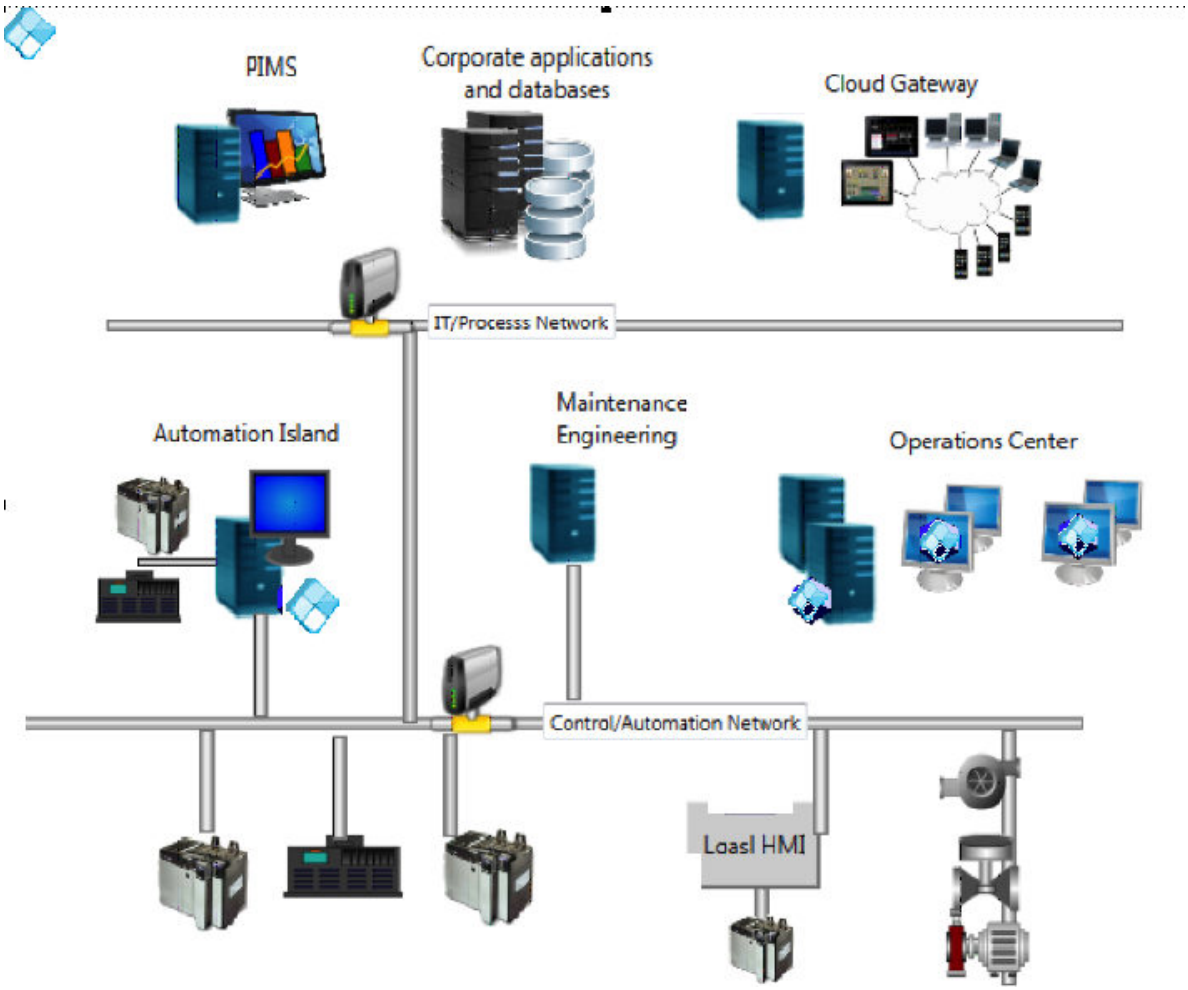


Topologia de Rede

O Action.NET pode ser aplicado em diferentes cenários e topologias de rede. As aplicações mais comuns, desde o chão de fábrica até a TI, incluem:

- Painel HMI local ou dispositivo embutido, funcionando de aquisição de dados por protocolos nativos e operação local.
- Servidor OPC e gateway de dados
- Supervisão e sistema SCADA em ilhas de automação
- Servidor Central para Centros de Operações e Controle Integrado de Salas
- Servidor de Dados prontos para "cloud" independente da fonte da camada de apresentação
- PMIS - painel em tempo real e Plant Information
- Gestão de Aplicações de Informações de Plantas.

O sistema Action.NET permite que os usuários foquem-se no o seu know-how e trabalhem nas soluções dos processo e aplicações, fornecendo componentes padronizados para as funções de software de infra-estrutura, tais como comunicação, gráficos e acesso de dados..



Arquitetura do Sistema

Banco de dados de tempo real

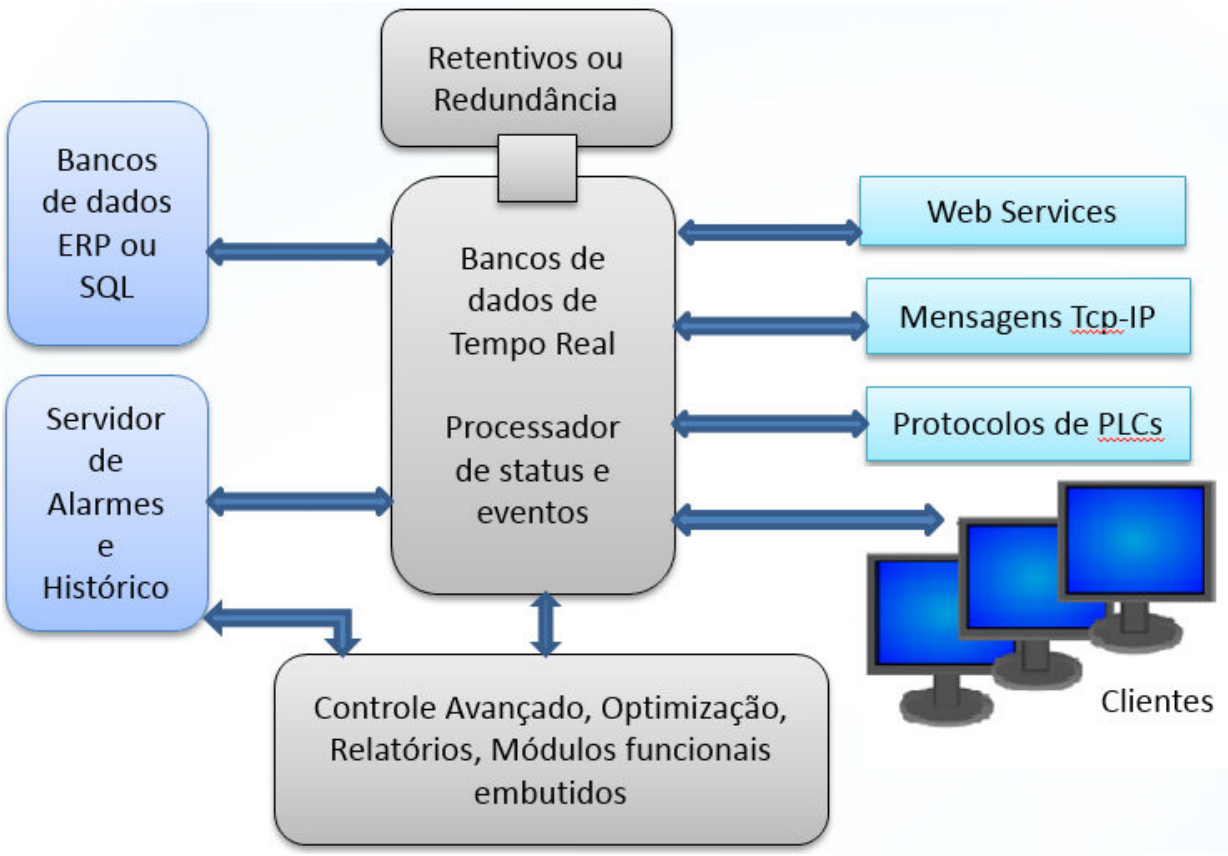
Um ponto chave de diferenciação da plataforma Action.NET é sua capacidade de ser aplicada tanto em complexos Controles de Processos críticos em tempo real, com as informações na memória como também nas aplicações de Nível 2 e Nível 3, onde as informações são armazenadas em bancos de dados SQL e ocorre também o intercâmbio de dados com aplicações externas.

O núcleo em tempo real de Action.NET é, uma base de dados controlada por eventos em memória. Esse banco de dados é o resultado de mais de 25 anos de aprendizado contínuo e é um projeto completamente novo, aproveitando as tecnologias atuais e as experiências passadas.

A especificação do projeto incluiu como critérios prioritários

- Confiabilidade e estabilidade operacional.

- A arquitetura distribuída, tanto internamente (mult-core CPUs) como externamente (vários usuários e projetos)
- Uso das plataformas operacionais para software e hardware de um amplo suporte e longevidade
- Aumentar a modularidade para reduzir os custos de implantação e manutenção.
- Combinar flexibilidade e interface de usuário fácil, permitindo o uso eficiente e a integração aberta com os sistemas e as necessidades futuras..



Gestão da Informação em Tempo real, TI e SCADA combinados

A função do banco de dados em tempo real, mostrado em cinza no diagrama, é a de permitir a modularidade do sistema, criando uma camada de abstração que permite isolar as tarefas para se comunicar com bancos de dados relacionais, comunicar com outros sistemas e com o campo, com a interface do usuário, e com sistemas de cálculo e de otimização. Sua estrutura permite a sincronização entre os vários processos independentes, de valores em tabelas em tempo real, notificação de eventos e atualização de informações.

Este aplicativo Manager utiliza componentes projetados especificamente para a plataforma NET com infra-estrutura adequada para o registro de eventos.

Fazendo uma comparação com uma supervisão em tempo real, ou de nível 1, temos as seguintes diferenças fundamentais:

Bancos de dados HMI / SCADA	Action.NET banco de dados em tempo real
Tipos básico de Tags, como boolean, int, float (real) e textos	Além destes, também gerencia eventos, datas, tabelas, consultas e dados estruturados.
Centrado em Servidor e processamento centralizado em um módulo	Processamento distribuído, multi-core e multi-processos, clientes e postos de controle avançados.
Interfaces de acesso proprietárias	Acesso através Interfaces classes .NET ou serviços web
Definição de variáveis em bancos de dados proprietários	Definição de variáveis em bancos de dados padrão com suporte para SQL e ADO.NET
Não permite hot swap (alteração da configuração do projeto, sem parar a execução da aplicação)	Permite atualizações "on-flight" com built-in de gerenciamento de versão.

Tabela 1: banco de dados em tempo real nova geração

Persistência e redundância

O acesso ao banco em tempo real para os valores originais e eventos em memória RAM, para a continuidade das operações em caso de falha crítica do sistema, pode ser usado tanto em um cache de disco (Persistência ou retentividade) ou uma sincronização TCP / IP com outra estação em uma arquitetura Hot-Standby. Ambas as soluções são built-in para o sistema e a melhor topologia depende dos requisitos específicos do projeto.

Engenharia e Runtime

Ao criar um aplicativo, a plataforma Action.NET usa o conceito de Engenharia e Runtime

Engenharia : Refere-se à configuração do aplicativo, ou a criação de um projeto Action.NET. As ferramentas de engenharia são um conjunto de utilitários de configuração, tais como editores de Tabelas, e ferramentas de desenho, que permitem implementar as especificações da aplicação em alto nível, utilizando interfaces de usuário de configuração amigável, em vez de programação de baixo nível. Quando uma lógica personalizada é necessária, interfaces de usuário permitem que você crie códigos nas linguagens C # ou VB.Net, de modo mais fácil e auto-organizado, mais uma vez proporcionando acesso em alto nível para os componentes do projeto.

Runtime : Refere-se a execução do Projeto configurado, o que significa executar o aplicativo desenvolvido. O sistema de tempo real executa a aquisição de dados, cálculo, registro e alarmes e fornece telas gráficas em tempo real, de acordo com a configuração do projeto de engenharia.

Modelo de Objetos e Namespaces

Mais avançado do que a maioria dos sistemas, onde você deve criar Tags ou variáveis para todas as propriedades internas e lógica personalizada para seus projetos, o Action.NET permite que suas aplicações acessem diretamente todos os objetos que foram criados em seu projeto. Isso significa que não é necessária a criação pelo usuário de Tags temporários para conseguir gerenciar o estado dos nós da rede de PLCs, ou o número total de alarmes em um grupo, ou o número de linhas em um conjunto de dados.

No Action.NET você pode acessar objetos “run time”, objetos de negócios (representando um nó de rede), um grupo de alarme ou conjunto de dados, e exibir as informações necessárias ou tomar medidas diretamente por meio de suas propriedades internas

Action.NET tem um bem fundamental modelo de objeto dot NET, 100% de código “managed”, visando especificamente o desenvolvimento de aplicativos de gerenciamento de dados em tempo real. O modelo de objeto hierárquico inclui os seguintes objetos de nível superior, que correspondem aos principais módulos em Action.NET:

• Tags	• Dataset
• Historiador	• Script
• Segurança	• Servidor
• Alarme	• Cliente
• Dispositivo	• Informações

Essa hierarquia de alto nível é implementada como. “Namespaces” do dot Net. Cada “namespace” tem as classes dot NET e objetos criados durante a configuração de projeto. Além de ter as definições de configuração, esses objetos também têm propriedades de tempo de execução, métodos e status.

Por exemplo o “namespace” Tag tem todos os tags na aplicação e cada tag possui propriedades “built-in” e propriedades do campo, tais como qualidade, TimeStamp, Min, Max, Unidades e muitos outros. Exemplos:

```
Tag.tagname1.bit0, tag.tagname2.timestamp
```

O mesmo conceito das propriedades dos Tags se aplica a todos os demais “Namespaces”, como por exemplo:

```
Alarm.TotalCount:, Alarm.Group.Warning.Disable:
```

Ao construir a configuração do projeto, preenchendo os campos de entrada ou a criação de scripts, o sistema sempre tem a facilidade de auto-completar Intellisense, que o orienta para as propriedades existentes possíveis de usar, no contexto, de acordo com o que você está editando. Este recurso permite que você facilmente escolha uma propriedade específica desejada.

Ao acessar um objeto do projeto no editor de scripts dot NET, é necessário prefixar o “Namespace” com o símbolo "@", a fim de evitar conflitos com nomes de variáveis locais dot NET. Por exemplo, nas “Tasks” de script e nos “CodeBehind”, deve-se usar:

```
@ Tag.Analog1
@ Device.Node.Node1.Status
```

O símbolo @ não é necessário em Grids e caixas de diálogo. Alguns campos de entrada podem exigir objetos de apenas um tipo, como Tag ou Display, o Intellisense irá guiá-lo automaticamente para os objetos permitidos.

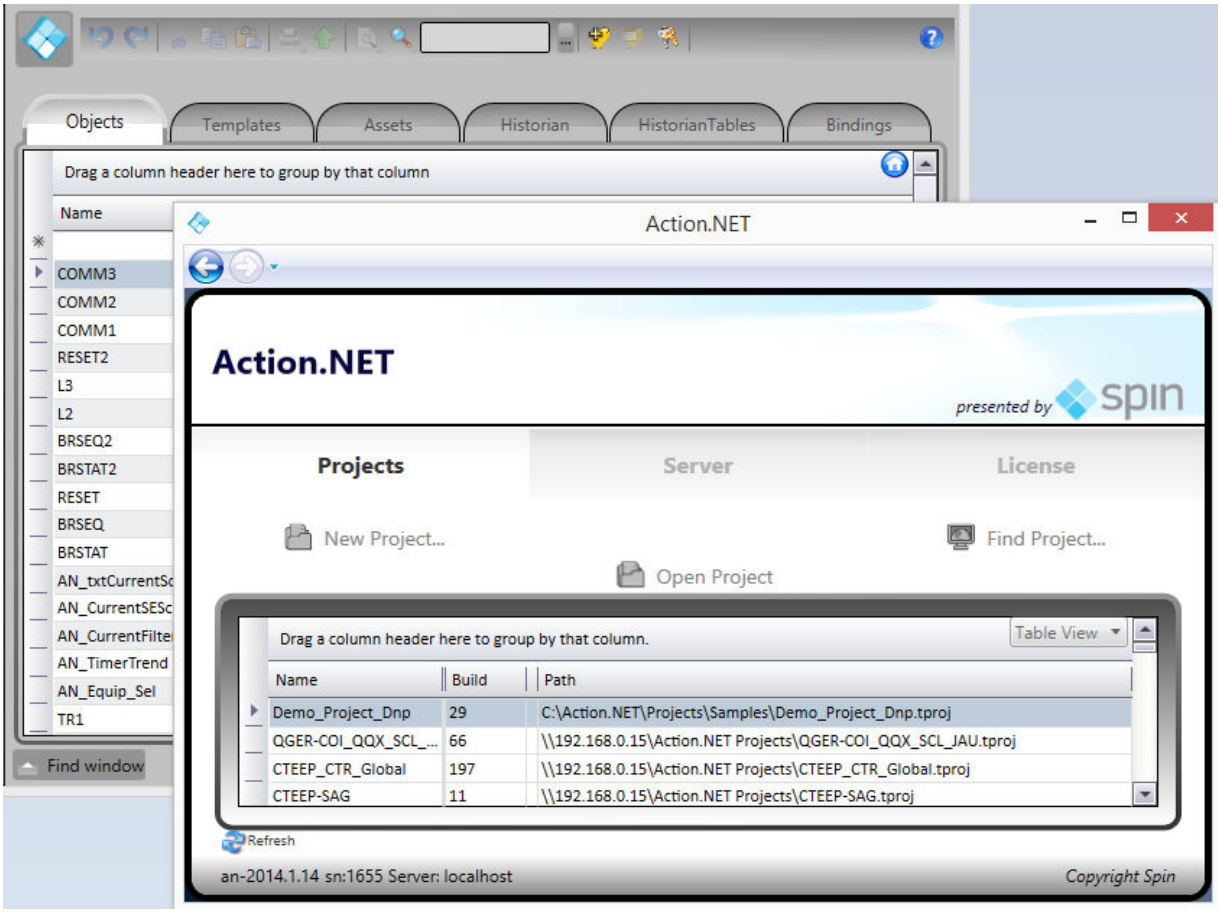
Para alguns usuários que não têm experiência anterior em dot NET ou em sistemas orientados a objetos similares, esses conceitos inicialmente são abstratos, mas quando melhor aprendidas as ferramentas de configuração de engenharia e os módulos Action.NET, o poder desses conceitos ficará bastante claro. O que é certo é que, quando o usuário se acostuma com os modelos de objetos e Intellisense, há um enorme incremento da produtividade e ele já não mais aceitará trabalhar em sistemas sem esses recursos.

Projetos Action.NET

O design inovador do Action.NET traz os conceitos de interface de usuário de ambientes Web e de "tablets" para as aplicações em desktop, tornando as ferramentas de engenharia mais fáceis, amigáveis e agradáveis de usar.

No Action.NET, múltiplos usuários podem trabalhar em um mesmo projeto simultaneamente. Um projeto Action.NET é um único arquivo em um computador, em um servidor remoto ou na nuvem!

]



Este desenho permite um acesso mais eficiente aos projetos localizados no computador local, estações remotas ou servidores na nuvem

O Action.NET cria automaticamente metadados sobre os projetos, com informações como pré-visualização de Tela, projeto e versão da ferramenta, data de modificação, modelo de projeto e muito mais. Você pode alterar o modo de visualização dos projetos para Cartões, ou Tabela, ou usar o botão Girar para alternar entre o ícone gráfico e um breve resumo do projeto.

O projeto tem uma configuração centrada em SQL, em que todas as definições de projeto, como telas gráficas, imagens gráficas, scripts, modelos de relatórios, estão armazenados em um único arquivo, que é um banco de dados SQL, cujo gerenciador está embutido no software, e fornecido com o Action.NET. Essa estrutura

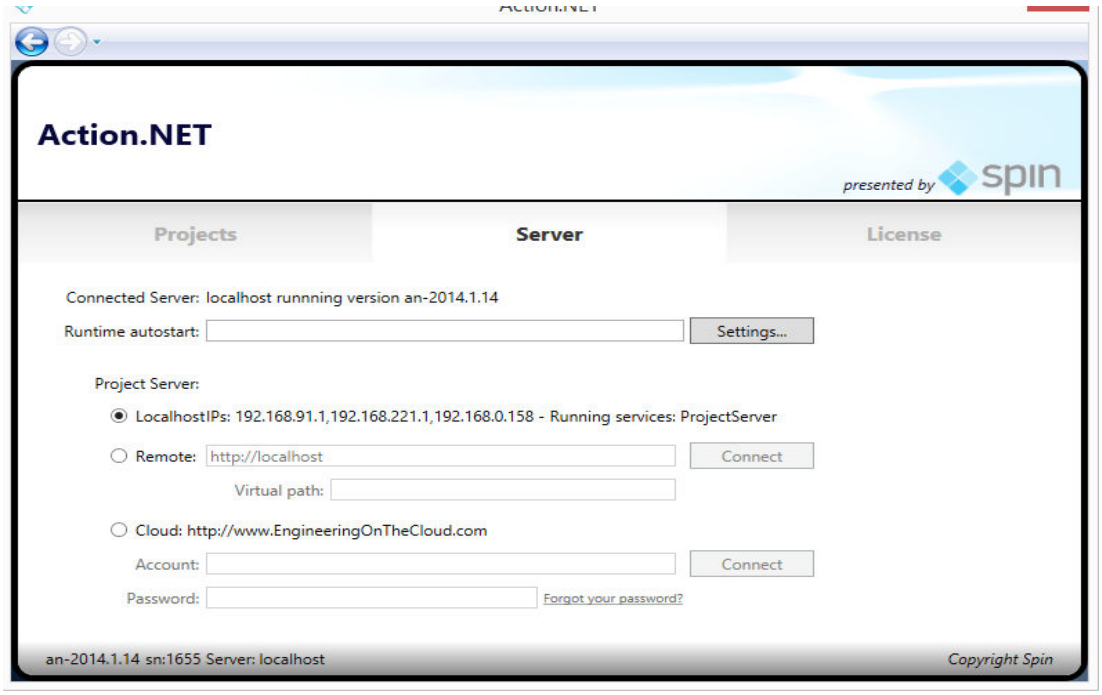
simples, facilita muito o back-up a movimentação e a implantação dos projetos. Também permite a edição de vários projetos ao mesmo tempo, assim, permite que vários usuários trabalhem simultaneamente.

No meio da tela você vê ícones para criar um "New Project", "Open Project" para abrir o projeto que você selecionou, e "Find Project" para definir caminhos padrão para localizar outros projetos.

Engenharia Distribuída

Na aba **Server** é feita a definição da localização onde você estará editando ou executando projetos.

Você pode editar e executar o projeto localmente (Localhost), a partir de um servidor remoto (Remote), ou de um servidor na nuvem(Cloud).



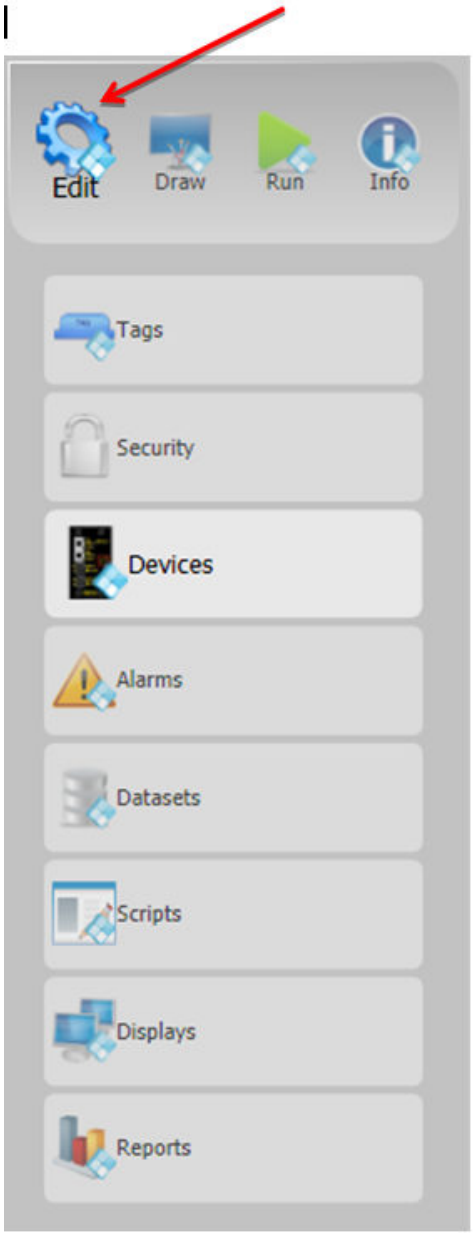
Todas as ferramentas e recursos de configuração são adequadas não importando se você está acessando um projeto a partir do computador local ou de um computador remoto. Não há necessidade de outros componentes de software para estas facilidades, o Action.NET fornece tudo que você precisa para a engenharia distribuída.

Configuração da área de trabalho

A idéia da interface de configuração é a de funcionar como uma "front-end web-page" onde você facilmente navega para formulários, preenchê os campos e os dados são salvos automaticamente para o Server do projeto, como um típico aplicativo web, sem a necessidade de abrir / salvar ou enviar periodicamente comandos de fechamento repetitivos. Para obter informações detalhadas sobre a área de trabalho refira-se a "Área de Trabalho de Engenharia" na pagina 59

Módulos Action.NET

O botão EDIT no espaço de trabalho do módulo de Engenharia permite o acesso as tarefas de configuração dos módulos Action.NET.



Para definir, verificar ou editar os "Tags" dos pontos a serem utilizados no projeto.

Para definir as permissões e níveis de acesso dos usuários do projeto.

Para configurar protocolos de comunicação e IEDs (equipamentos de comunicação e de aquisição de dados) que serão utilizados pelo projeto

Para configurar "Tags" de alarme e Grupos de Alarmes para a geração de alarmes sob condições definidas.

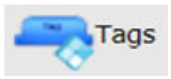
Definição de fontes de dados como arquivos bancos de dados, tabelas específicas e "queries" pré-definidas.

Para configuração de "scripts" que poderão ser em C-Sharp ou VB.Net, (em dot NET) compilados e ligados dinamicamente antes da execução dos projetos.

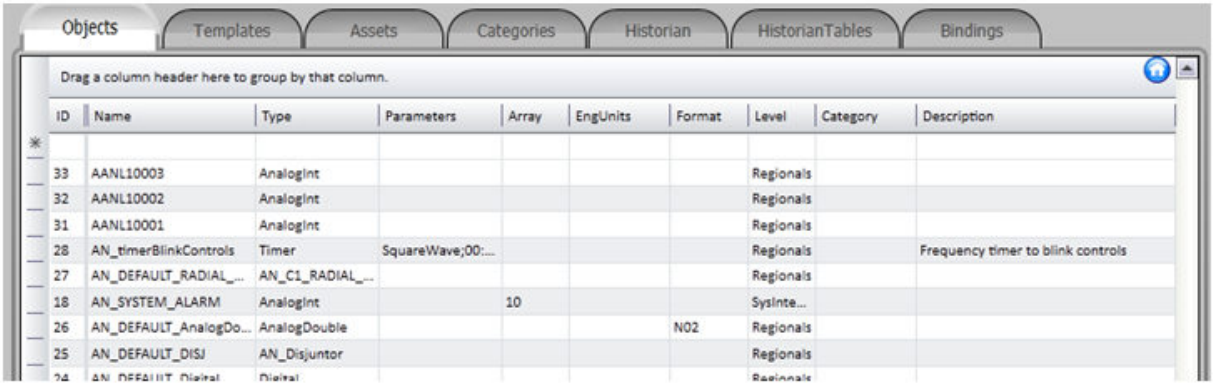
Para configurar ou selecionar telas gráficas dinâmicas que vão mostrar em tempo real o estado dos processos supervisionados

Para configurar Relatórios, que podem incluir símbolos gráficos atualizados, em formato XPS, HTML, ASCII e Unicode.

Tags de tempo real



O banco de dados em tempo real Define as variáveis do processo de aplicação (Tags), tipos complexos (Templates) e Níveis hierárquicos (Assets). O banco de dados em tempo real assegura, sem necessidade de qualquer programação adicional, a sincronização de dados entre os vários processos do servidor e das várias estações cliente. Um conjunto de propriedades pré-definidas, como a qualidade dos dados, estampa de tempo, estado de bloqueio e valor bloqueado, simplifica a modelagem de sistemas. Grupos de tags e outros componentes do projeto pode ser atribuído a níveis hierárquicos (Assets), criando uma visão de ativos de seu sistema, como por exemplo: Regional, Sub-estação, Setor de tensão, etc..



O banco de dados em tempo real também tem suporte para matrizes tri-dimensionais com tamanho especificado de modo dinâmico, permitindo uso eficiente de memória, e estruturas e modelos definidos pelo usuário. O Action.NET tem o mais completo conjunto de entidades em tempo real. Foi a primeira plataforma que implementa a correspondência completa entre Tags de tempo real, variáveis em dot NET e campos de bancos de dados SQL, como mostra a tabela a seguir.

Tipo do tag Action.NET	Tipo em .NET Framework	Tipo em database SQL
Digital	Boolean	Bit
Integer	int32	int
Long	int64	bigint
Double	double	float
Decimal	decimal	decimal
Text	string char[]	nchar, ntext, char, text, varchar
DateTime	DateTimeOffset	DateTime, DateTimeOffset
TimeSpan	TimeSpan	Time
Timer	int32	int
Image	byte[]	Binary, Image, VarBinary

Tipo do tag Action.NET	Tipo em .NET Framework	Tipo em database SQL
Guid	Guid	UniqueIdentifier
DataTable	System.Data.DataTable	

Telas Gráficas

O dot NET Framework., entre os vários avanços da tecnologia, trouxe também o Windows Presentation Foundation (WPF) que é a tecnologia gráfica que permite fácil acesso a todo o potencial das modernas placas de vídeo e aceleradores de hardware para gerenciamento de imagens. Outro benefício foi a introdução do XAML (Extended Markup Language) que permite criar a interface gráfica do usuário de forma declarativa, possibilitando a construção de interfaces com mais funcionalidades, em menos tempo de programação.

O Action.NET foi inteiramente desenvolvido sobre as facilidades disponibilizadas pelo WPF e XML. Tanto as ferramentas de engenharia, que proporcionam interfaces muito amigáveis, como os módulos de tempo real de visualização, que facilitam aos engenheiros criar aplicações avançadas estado-da-arte, incluindo acesso a todo o potencial de placas de vídeo atualmente disponíveis, com mais rapidez e eficácia no desenvolvimento.

O Action.NET inclui um poderoso editor de recursos gráficos para criar as aplicações de tempo real, exclusivamente criado para proporcionar a melhor integração com os outros módulos e com o banco de dados em tempo real. O editor inclui também um conjunto de animações dinâmicas para os objetos de visualização e uma biblioteca de símbolos, que pode ser estendida pelo usuário.

Cinco tipos de clientes remotos são suportados

Rich Clients: é executado como uma aplicação desktop, permite o bloqueio da alternância entre esta tarefa e outras do Windows. Este tipo é ideal para aplicações de controle de processos que necessitam de alto desempenho, disponibilidade e segurança.

Smart Clients: Usa a tecnologia "Clique-Once", que instala e atualiza o software em clientes remotos com um único clique a partir de um navegador e ainda é automaticamente atualizado nos clientes remotos quando for atualizado no servidor. Este tipo usa todo o poder do computador remoto porém ainda mantendo as vantagens de um sistema centralizado de instalação.

Clientes da Web: Executa diretamente da partir do Web-Browser, sem a necessitar a instalação de qualquer software (nem quaisquer controles Active-X!). A segurança "partial-trust" garante que é executado em um ambiente seguro e completamente isolado.

Thin Clients: Executa usando a conexão de desktop remoto ou servidor de terminal ou outra tecnologia de thin client. Normalmente usado para rodar em tablets que usam Android ou WinRT.

Tablets, app nativo do iPad . As mesmas telas criadas em computadores Windows são automaticamente geradas para rodar nativamente em iPads em iPhones, combinando a experiência do usuário e a segurança de um aplicativo nativo com os benefícios de ter apenas a configuração do projeto.

Veja mais em "Telas e Símbolos" na pagina 147.

Segurança e Redundância

No Action.NET, Security é usado para definir quais usuários têm acesso e como aos vários componentes do projeto. Nisto se inclui a capacidade de bloquear ou permitir o usuário de alterar os valores do Tag, executar comandos, abrir Displays, gerar relatórios e muito mais.

Sessões de usuários podem ser controladas de modo que sejam automaticamente encerradas, baseado em inatividade, a duração total da sessão, ou ambos.

As condições on-line de alarme e de segurança são automaticamente replicadas em projetos com redundância.

Juntamente com os recursos de alarmes e eventos, as ferramentas de segurança fornecem todas as ferramentas necessárias para criar aplicativos compatíveis com as recomendações FDA CFR 21 Parte 11.

Veja mais em "Segurança e Usuários" na pagina 97.

Tendência e Historiador

O Módulo Historiador possibilita ao usuário facilmente configurar o registro de variáveis do processo em bancos de dados SQL. Um gerenciador banco de dados SQL built-in é fornecido para bases de dados de pequeno e médio porte. Grandes sistemas podem ser configurados para usar o Microsoft SQL Server, Oracle ou qualquer banco de dados compatível com OLE-DB, ODBC ou ADO.NET.

O Action.NET também tem a opção de usar o sistema OSIsoft PI para o servidor historiador, sem necessidade de programação, incluindo completa integração e acesso transparente.

As ferramentas gráficas incluem um objeto gerador de gráfico de tendência poderoso com suporte para informações de qualidade, anotações, gráficos verticais, comparação de series e muito mais.

Veja mais no "Historiador e Registro de dados" na pagina 91

Devices - Comunicação com IEDs

Este módulo gerencia a comunicação com IEDs, controladores programáveis ou outros equipamentos de campo.

O Action.NET possui incorporados módulos de protocolo de comunicação para acessar diretamente PLCs, IO Remoto, padrões Fieldbus, single e multi-loops, scanners, códigos de barras, dispositivos de RFID e displays digitais. OPC DA também é suportado, mas na maioria dos casos não é necessário: os protocolos nativos são usados quando disponíveis, pois possibilitam melhor desempenho, mais fácil configuração e melhor controle de mensagens. Reserva-se o uso de OPC para quando um driver nativo não está disponível ou quando a especificação da arquitetura assim o exige.

A Ferramenta de Configuração de Devices pode importar bancos de dados de servidores OPC, CSV ou arquivos de texto. Se o dispositivo for compatível, ele implementa automaticamente "multi-threading" em redes TCP / IP. A sintaxe de endereçamento segue a convenção de nomenclatura do dispositivo remoto, tornando a configuração e manutenção muito mais fácil, um conjunto completo de monitores de desempenho e ferramentas de diagnóstico está incluído.

Veja mais no "Dispositivos (Devices) e Interfaces" na pagina 103

Alarmes e Eventos

O Sistema de alarmes do Action.NET prevê uma enorme flexibilidade no gerenciamento de alarmes da aplicação. Vários níveis de alarme para cada ponto / tag podem ser definidos e toda uma gama de comportamentos, tais como o reconhecimento, a Tela visual e sonora, o registro em base de dados, são características Bult-in para simplificar a configuração.

As condições de alarme são automaticamente replicadas em servidores redundantes, certificando-se que não há alarmes perdidos em caso de uma falha do computador. As ferramentas de alarmes e eventos também fazem parte das ferramentas necessárias para criar aplicativos compatíveis com a recomendação FDA CFR 21 Parte 11.

Veja mais em "Alarmes, eventos e Audit Trail" na pagina 117

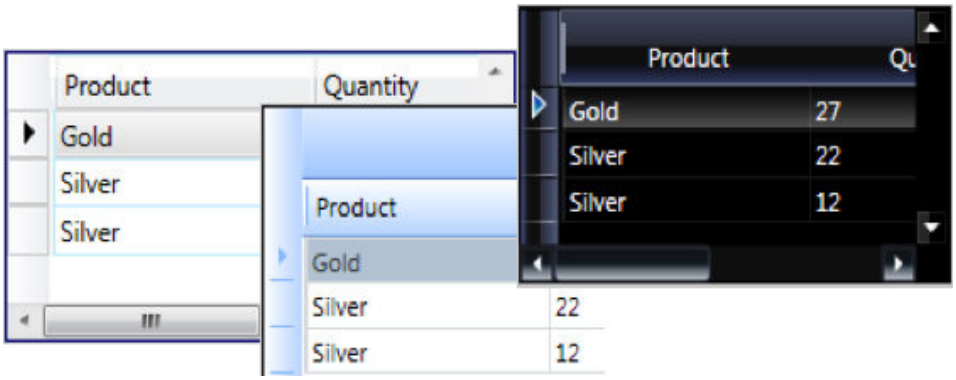
Datasets e SQL

O Módulo Dataset incluído no Action.NET fornece uma interface fácil de usar para trocar dados em tempo real, com bancos de dados externos, XML, CSV ou arquivos de texto e consultas SQL e acesso a tabelas.

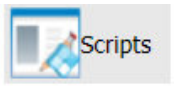
Para os bancos de dados e fontes de dados mais comuns (Microsoft SQL Server, Oracle, arquivos CSV, Microsoft Access, PI, Firebird, Informix, Excel), o Action.NET fornece configurações pré-definidas que reduzem a configuração para alguns cliques do mouse. Qualquer banco de dados que suporta ODBC, ADO.NET ou OLE-DB pode ser acessado. Um gerenciador de banco de dados SQL embutido também é fornecido como um banco de dados local para a sua aplicação (s).

Os dados coletados com os Datasets podem ser mapeados dinamicamente em pontos / tags de tempo real e podem ser utilizados em scripts ou relatórios, ou apresentados em telas usando um poderoso Data Grid Visual Object, com editores de células e suporte para vários temas de visualização.

Veja mais em "Datasets e Intercâmbio de dados" na pagina 127



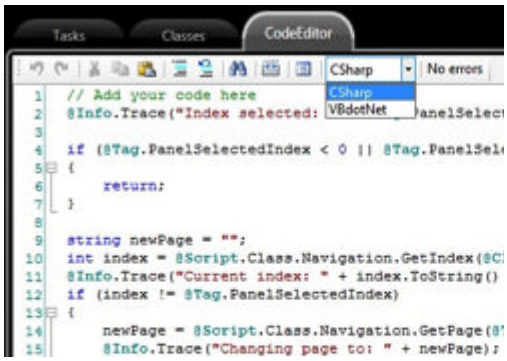
Scripts e dot NET



O Action.NET suporta completamente o Microsoft dot NET Framework em um ambiente integrado embutido. Dentro da arquitetura do Action.NET você pode compilar, obter referência cruzada dos objetos e acessar diretamente (usando o Intellisense) as classes dot NET e seus objetos do projeto, incluindo Alarmes, Relatórios e nós de comunicação.

As linguagens dot NET fornecem uma opção mais poderosa e segura quando comparada com VBA ou VBScript, que são interpretados e levam a que muitos erros só possam ser encontrados ao executar o seu projeto. Com o VBA ou VBScript, já com o projeto em produção, muitas vezes você ainda tem erros de execução, com consequências indesejáveis. O ambiente "managed" do Microsoft dot Net Framework fornece suporte para descobrir e se recuperar de exceções, proporcionando assim um ambiente altamente confiável para o sistema de execução em tempo real. Os scripts de projetos e lógica de negócios podem ser escritos em C # ou VB.NET. Um sistema de depuração completa está incluído, possibilitando colocação de pontos de interrupção, execução passo a passo, e mudanças de código on-line durante a depuração . Há também um conversor de linguagem embutido, que lhe permite mudar o código criado dinamicamente entre o C # e VB.NET.

Otimização e controle avançado

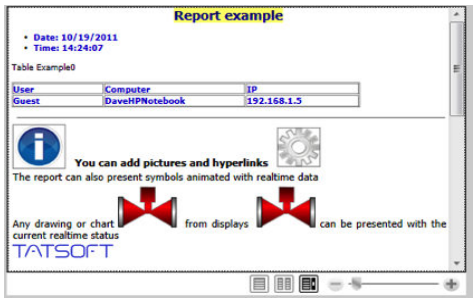


Apesar de usarmos o nome "script" para apresentar uma nomenclatura familiar para os novos usuários, este módulo do Action.NET é um "engine" dot NET, que suporta multi-threading e programação avançada. Além das lógicas de interface do usuário, ele pode trabalhar como um módulo de otimização de processos, realizando o cálculo de modelos matemáticos em um ambiente seguro e de alto desempenho. O desenvolvimento destes módulos vai se concentrar apenas em algoritmos e processos, uma vez que as interfaces para valores de leitura e dados históricos, disparo de eventos e escrita de valores são funções padrão e componentes já fornecido pelo Action.NET

Veja mais em "Scripts e .NET Framework" na pagina 137

Relatórios e acesso a dados

Action.NET suporta acesso SQL, Web-Services, XML e outras interfaces de intercâmbio de dados para fornecer dados para ferramentas de relatórios externos. Além disso, ele tem o seu próprio Editor do Relatório.

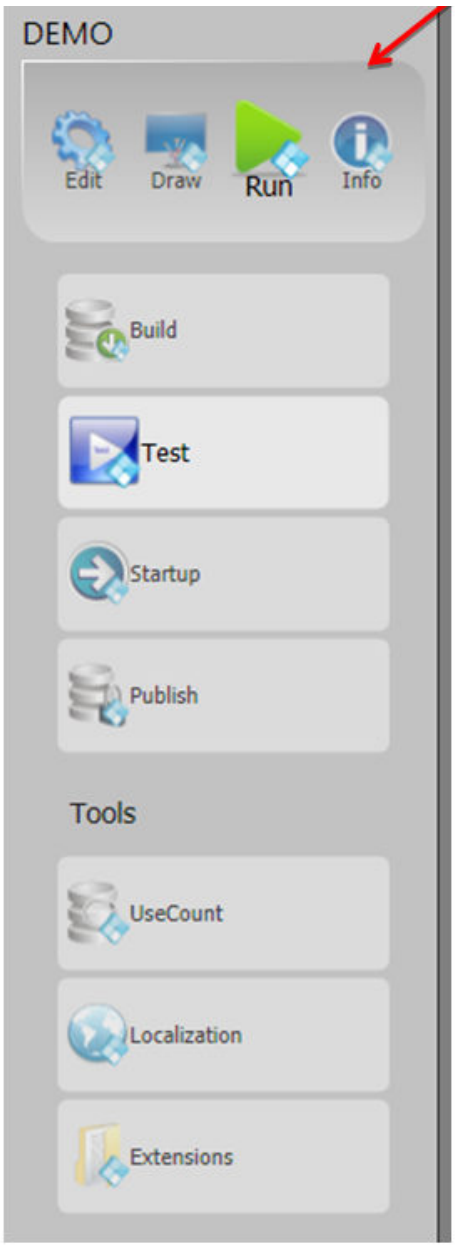


O Editor de Relatório permite a inclusão de texto dinâmico, símbolos gráficos dinâmicos e gráficos, conjuntos de dados e os resultados de uma consulta, em uma funcionalidade de edição fácil de usar. Os relatórios podem ser salvos em HTML, texto, XPS ou PDF e facilmente ser apresentados em clientes remotos e em telas na web. Um controle de navegador nativo para visualização de relatórios está incluído, por isso não há necessidade de programas de terceiros para ver gerado relatórios anteriormente configurados.

Veja mais em "Relatórios e Acesso a dados" na pagina 193.

Testando e executando projetos

O ambiente de execução fornece acesso a todos os recursos de execução do projeto.



Build deve ser utilizado para solicitar a preparação do Projeto para execução em tempo real, e o salvamento da história do projeto, incluindo compilações e "builds" de scripts em bibliotecas de código executável.

Teste para testar o projeto e verificar sua execução em tempo real antes de publicá-lo para uso no Servidor real.

StartUp Permite especificar ferramentas de diagnose e os módulos a serem utilizados para testar aplicativos e scripts em execuções do projeto

Publish - Publicar o projeto para obter o arquivo final de instalação no servidor. Obtém-se um novo identificador da Revisão Publicada do projeto.

UseCount - Apresenta um sumário do projeto como numero de Tags utilizados. Fornece também um relatório de referência cruzada dos objetos do projeto.

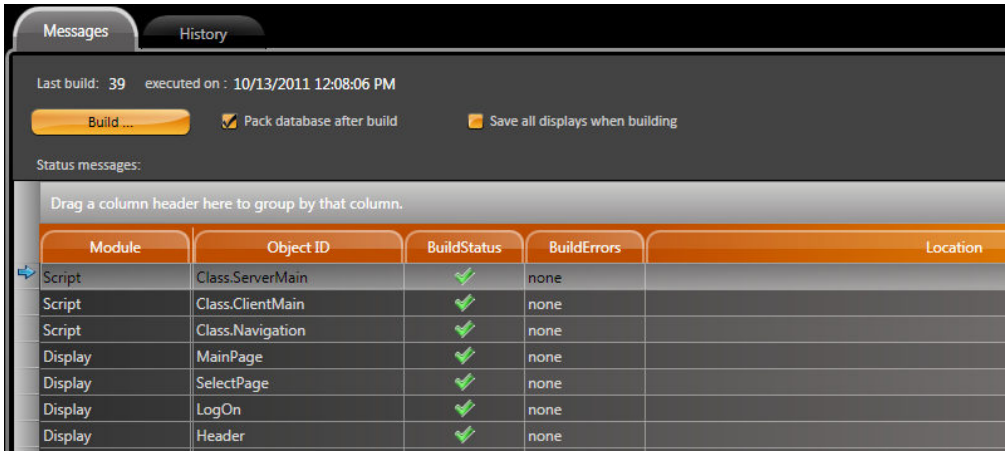
Localization - Para a customização do idioma a ser utilizado nas mensagens de alarmes e textos de telas do projeto.

Extensions - Importar dados de configuração já criados para Servidores OPC, de quaisquer tabelas de configuração em formato CSV, ou mesmo de um projeto completo. Configurar o uso de aplicativos externos (Modelos) compartilhando dados em tempo real.

Building

Build serve para recompilar todos os objetos telas e scripts do projeto para verificação final quanto a possíveis erros antes da colocação do projeto para a produção.

É feita uma verificação completa da lógica da aplicação durante a preparação dos arquivos para a versão final de produção. Esta operação não é necessária durante o desenvolvimento do projeto, já que todas as modificações são automaticamente e transparentemente sendo compiladas em "background" enquanto as edições são feitas.

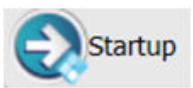


Testando o projeto

Esta função leva-o a executar o projeto no modo de *Teste,* o que significa habilitar as proteções que permitem a execução, mesmo na mesma máquina que o projeto está executando na partida. As funções de gravação do *Historiador* e de *Alarmes* são direcionadas para utilizar arquivos temporários ao invés dos definidos na base de dados do projeto final em produção.

Opcionalmente, pode-se habilitar ferramentas de diagnose para auxiliar a resolver alguns problemas que porventura possam estar presentes na construção do projeto. Pode-se escolher desabilitar qualquer dos módulos durante o teste.

Startup do Projeto



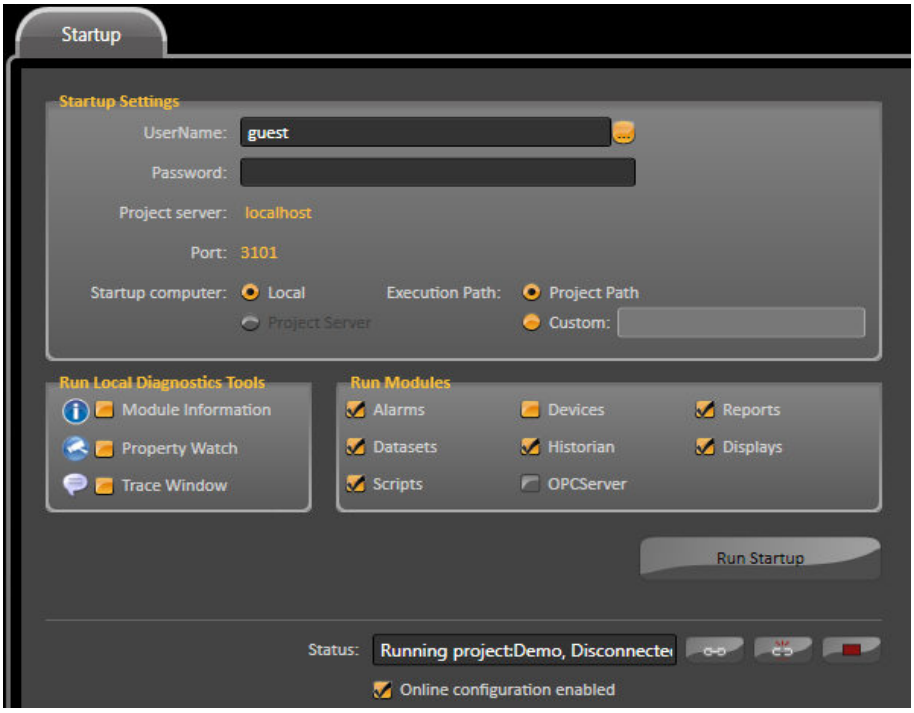
Esta área permite a configuração de como o projeto será executado em modo final de produção. Significa habilitar as funções de gravação de *Historiador e Alarmes* para utilizar as bases de dados reais definidas no projeto ao invés de arquivos temporários utilizados no modo de teste.

Opcionalmente pode-se habilitar as ferramentas de diagnóstico para auxiliar na solução de algum problema que possa estar ocorrendo no projeto. Pode-se escolher desabilitar qualquer dos módulos durante o teste. O check-box *"Configuração On-line"* quando marcado, indica que deve ser aceita a possibilidade de alteração de configurações durante a execução do projeto. Assim durante a execução, se forem feitas

alterações nos parâmetros do projeto, as mesmas serão efetivadas e incorporadas ao projeto enquanto executando.

Opcionalmente pode-se habilitar as ferramentas de diagnóstico para auxiliar na solução de algum problema que possa estar ocorrendo no projeto. Pode-se escolher desabilitar qualquer dos módulos durante o teste.

O check-box *"Configuração On-line"* quando marcado, indica que deve ser aceita a possibilidade de alteração de configurações durante a execução do projeto. Assim durante a execução, se forem feitas alterações nos parâmetros do projeto, as mesmas serão efetivadas e incorporadas ao projeto enquanto executando.



Publicando o Projeto

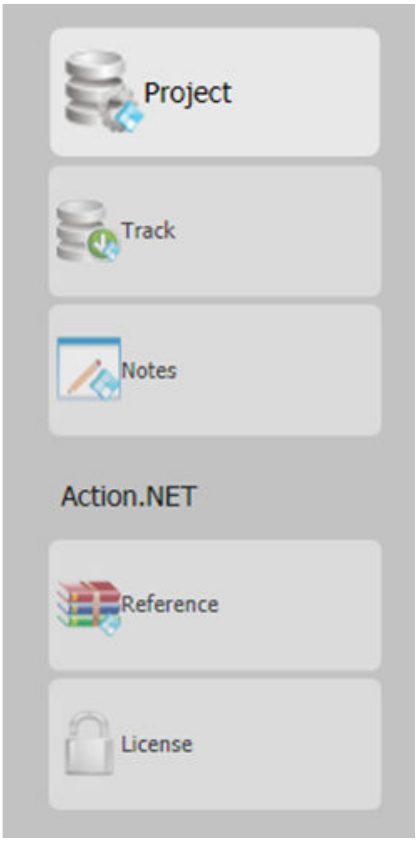
A função *Publicação* cria uma versão para o projeto protegida contra a escrita (read-only) em um arquivo apropriado para ser instalado nas máquinas definitivas de produção.

Quando o comando de *Publicação* é executado, um novo arquivo de projeto (com a extensão **.teng**) é criado, utilizando para numero de versão o que estiver especificado no campo correspondente desta área, mostrada a seguir. Os projetos Publicados (arquivos com a extensão **"tproj"**) são em tudo similares aos projetos correntes em desenvolvimento (extensão "**proj**") mas somente podem ser abertos em modo protegido (read-only). Esta funcionalidade permite que se tenha uma versão segura de aplicações publicadas que pode ficar como "backup" na própria instalação final.

Veja mais em "Execução, Testes e Publicação de Projetos" na página 195

Informação e Controle de Versão

O ambiente de Informações fornece acesso a detalhes de configuração atual do projeto



Mostra as versões do produto e do projeto, a localização do projeto, a linguagem utilizada e vários outros detalhes .

Para se obter informações sobre as Publicações do projeto, números de versões e história de alterações realizadas.

Para preenchimento de notas e observações importantes para outros usuários que estejam desenvolvendo o projeto.

Para chamar o Auxilio On-line de Referência do Action.NET

Mostra informações específicas do Licenciamento corrente do produto. Uma aba de Registro facilita o registro on-line do software.

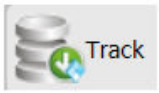
Verificando informação sobre o Projeto

O próprio **Action.NET** ajuda na obtenção e na guarda de informações sobre o projeto.

Durante o gerenciamento das versões de projetos esteja-se desenvolvendo, instalando, comissionando ou mantendo um projeto, é importante ter conhecimento da versão do software sendo utilizado, dos detalhes da licença, do projeto e de outras informações de configuração.

A abas componentes da área de trabalho do *Projeto* mostram estas informações e permitem a sua alteração se necessário. Alguns destes atributos estão disponíveis também em tempo real se necessitadas.

Controlando alterações de projeto



Uma funcionalidade muito importante disponível no **Action.NET** é a possibilidade de manter o rastreamento de todas as alterações feitas no projeto. A rastreabilidade auxilia na verificação das configurações do projeto em vários estágios de seu desenvolvimento e implementação. Isto é importante porque, muitas vezes, se precisa ser capaz de verificar qual a versão do projeto que você está usando, quando desenvolvendo aperfeiçoamentos e trabalhando com diagnósticos.

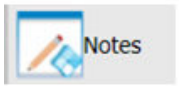
O **Action.NET** rastreia as alterações de todas as partes de configuração do projeto.

A aba *Tabelas* mostra uma lista de todas as tabelas do projeto que tiveram algum campo alterado, com a indicação linha que foi alterada Com um duplo clique sobre a linha se vai para o editor (datagrid, ou editor de código ou de telas) que gerou a mudança.

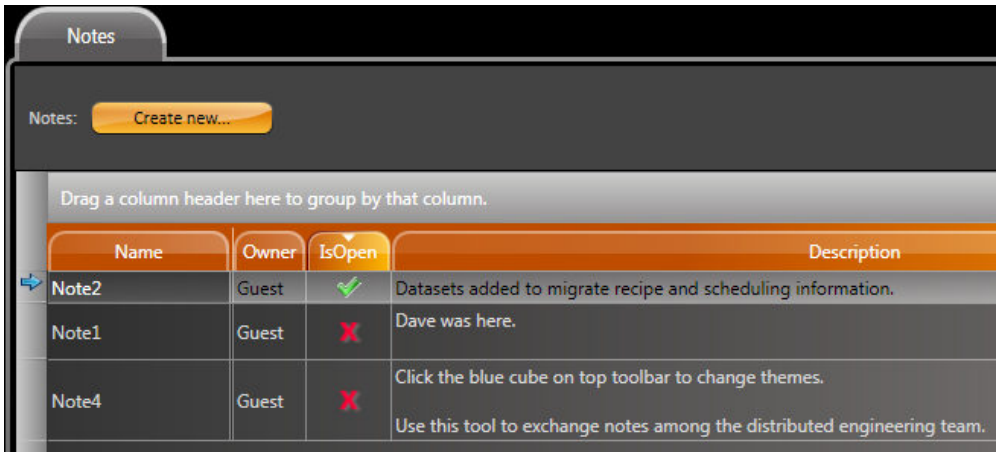
A aba **Alterações (Changes)** fornece informações sobre o módulo específico em que as alterações ocorreram, a alteração específica que foi feita, a operação de Build específica em que a alteração ocorreu, bem como a data da alteração e quaisquer comentários feitos pelo usuário.

A aba **Releases** fornece informações sobre a data de quando o projeto foi publicado, que o publicou, e o numero do ultimo Build em que o projeto foi publicado.

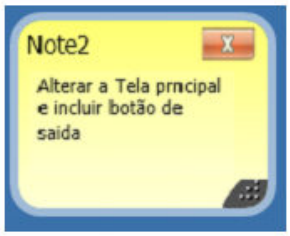
Adicionando e compartilhando anotações de projeto



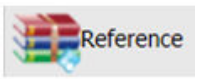
Muitas vezes durante o desenvolvimento e implementação de um projeto, os desenvolvedores necessitam ser capazes de deixar notas para si mesmos, ou para outros que estejam colaborando no projeto, de forma a, principalmente, documentar as ações e motivos de decisões tomadas. No Action.Net isto pode ser feito pelo uso do utilitário **Notes** disponível no software.



As Anotações criadas são do tipo “post-it” e são visíveis por todos os membros do time de desenvolvimento do projeto. As Anotações ficam visíveis no “desktop” quando se está editando o projeto.



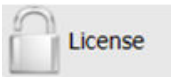
Referência



O *Action.NET* inclui um guia de Referência (auxilio on-line) para explicar as ferramentas de desenvolvimento. Use o índice para pesquisar palavras –chave on-line.

O sinal de interrogação no canto superior direito do Configurador, também chama o mesmo auxilio on-line.

Informação de Licenciamento



A aba de *Licenciamento* apresenta informação sobre a licença atualmente instalada para o *Action.Net* , nesta máquina. Esta informação será de grande utilidade se for necessário contatar a área de suporte técnico da Spin Engenharia e quando for época de fazer atualização para uma versão mais nova do software..

Veja mais em "Versões e Licenças Action.NET" na página 217

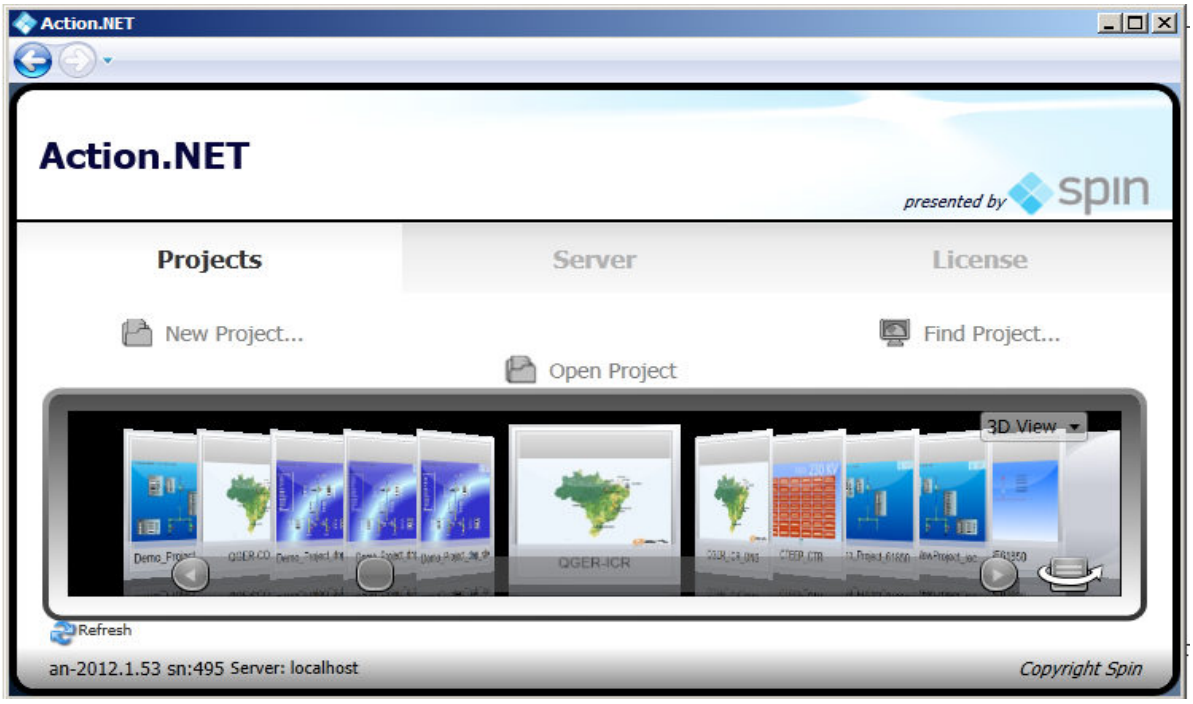
Gerenciamento de Projetos

As seções a seguir descrevem como criar e gerenciar projetos:

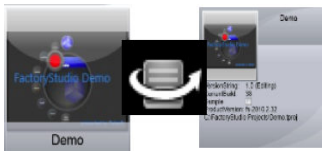
- "Visualizando Projetos" na pagina 47
- "Criando um Projeto" na pagina 49
- "Encontrando Projetos" na pagina 50
- "Permitindo o acesso remoto" na pagina 52
- "Accessando Projetos remotamente" na pagina 53
- "Executando Projetos" na pagina 54
- "Atualizando projetos" na pagina 54
- "Configuração de projeto" na pagina 55

Visualizando Projetos

Após iniciado o aplicativo Action.NET (Veja "Iniciando o Action.NET" na pagina 13), pode-se ver a janela de Gerenciamento de projetos do sistema, mostrada abaixo.



Os projetos apresentados na lista são dependem do Project Server selecionado (definido na aba Servidor) e os caminhos de pesquisa selecionados (definidos com o botão Find projeto).

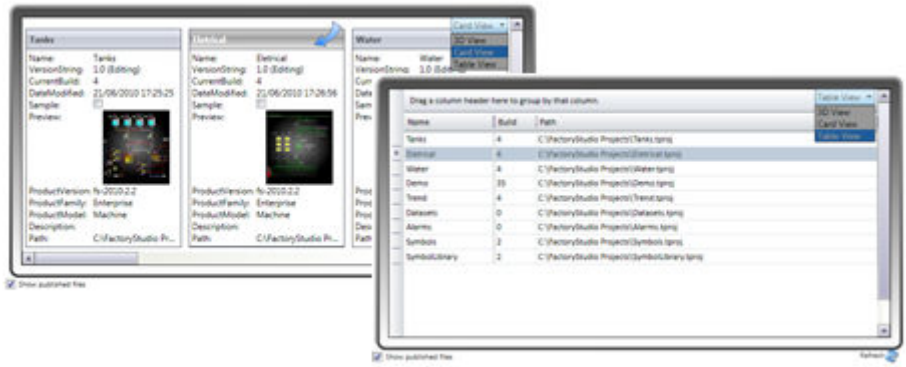


O Action.NET cria automaticamente metadados sobre os projetos, com informações como pré-visualização de Tela, projeto e versão da ferramenta, data de modificação, modelo de projeto e muito mais. Usando a combo-box a direita você pode alterar o modo de visualização dos projetos para Cartões, ou Tabela, ou usar o botão Girar para alternar entre o ícone gráfico e um breve resumo do projeto.

Veja "Configuração de projeto" na pagina 55 para obter detalhes sobre as informações de metadados do projeto.

Um Projeto em Action.NET contém toda a configuração para uma aplicação específica. Esta configuração do projeto é armazenada em um único arquivo com extensão **tproj**.

Dica: Se a lista não está mostrando nenhum projeto, é porque o sistema não foi capaz de localizar arquivos de projeto nos caminhos do servidor e de pesquisa selecionados. Use o "Find Projects" para pesquisar projetos no computador local, use a aba "Servidor" para se conectar a um novo Project Server, ou use o "Create Project" para criar um novo projeto.



Os Projetos, mostrados nesta lista, são todos os encontrados nos caminhos de pesquisa, que podem ser definidos usando o comando **Find Projects**. Consulte "Encontrando Projetos" na pagina 50.

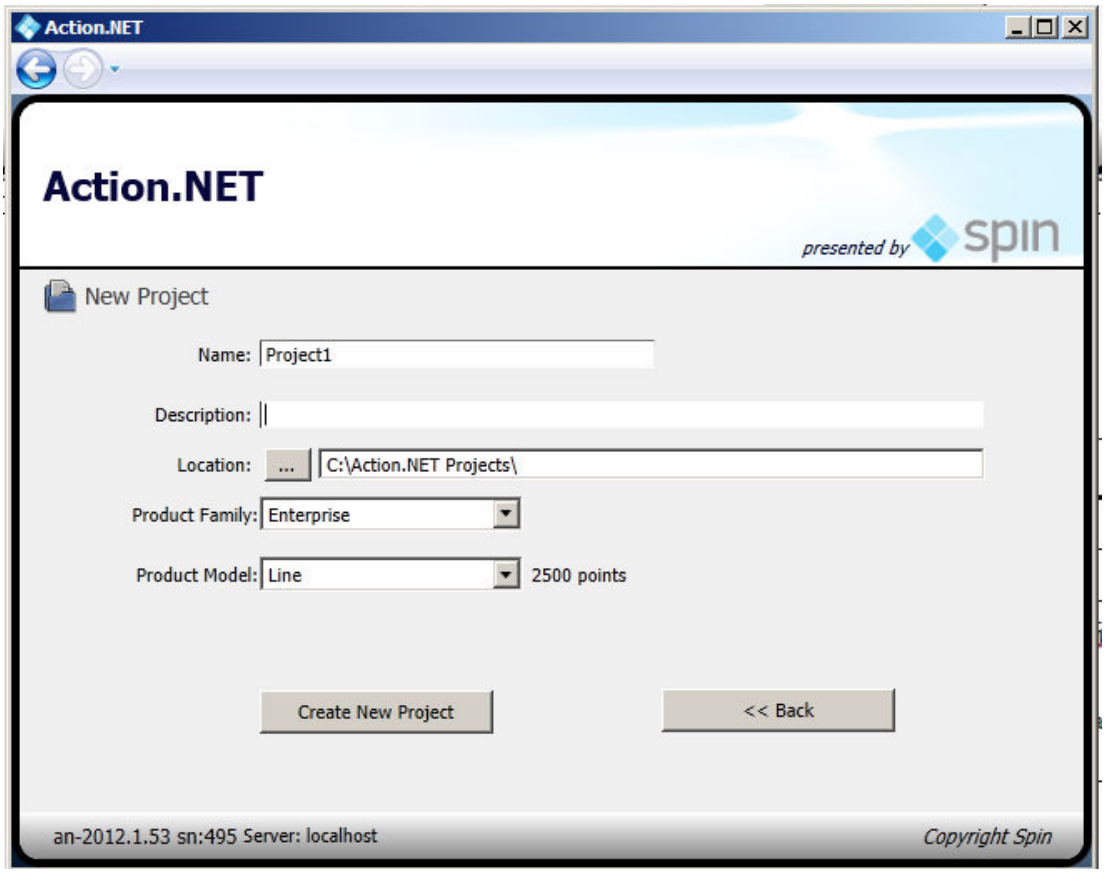
Projetos localizados em computadores remotos ou em servidores na nuvem podem também ser gerenciados usando a configuração do servidor, consulte "Accessando Projetos remotamente" na pagina 53.

Projetos retém informações sobre a versão do Action.NET que foi usado para criar o projeto original. A menos que você atualizar o projeto para uma nova versão, o projeto abre sempre usando a versão com a qual ele foi criado. Para mais informações, consulte "Atualizando projetos" na pagina 54.

Criando um Projeto

Clicando em **New Project** é apresentada a tela abaixo com campos para a especificação de atributos do novo projeto a ser criado. Para cria-lo preencha os campos com o nome e a descrição do novo projeto e escolha os atributos de produto, modelo, idioma e linguagem a ser utilizado em "scripts". Ao final pressione o botão "Criar novo projeto" para que seja, de fato, criado um projeto ou clique em "Voltar" se desistir criar um novo projeto.

.



Para criar um projeto:

1. Clique em **New Project**.
   - Aparece a figura com a janela de criação de projetos mostrada acima.
2. Entre ou selecione a informação conforme você precisa:

Campo	Descrição
Name	Entre um nome para o projeto
Description	(opcional) Entre com um texto de descrição do projeto

Campo	Descrição
Location	Clique no botão (...) para selecionar ou criar uma nova pasta para o projeto Nota: Você pode configurar o acesso remoto aos projetos com base em diretórios (pastas). Certifique-se de colocar projetos locais em uma pasta diferente de projetos que você permite o acesso remoto.
Product Family	Dependendo da sua licença Action.NET, você pode ter aqui as opções para construir aplicações para uma versão específica do Action.NET. **Dica:** Se você não tem uma licença, só a família Express será mostrada. Mesmo quando você tem uma licença você pode querer criar projetos tipo Expresso para compartilhar demos e tutoriais com os usuários que não possuem licenças instaladas. • Veja "Versões e Licenças Action.NET" na pagina 217, para mais informações sobre a Família do produto. Uma descrição básica da família é apresentada a seguir: • **Enterprise**—Funcionalidade completa do Action.NET • **HMI**—Use para aplicações menores e sistemas embarcados. • **Express**—Use somente para fins de teste ou demonstração. • **OPCServer**—Criar um projeto para trabalhar apenas como um servidor OPC, fornecendo dados para outras aplicações a partir dos drivers de protocolo nativos • Você pode alterar essa configuração mais tarde, se necessário.
Product Model	Dependendo da sua licença Action.NET, você pode ter opções aqui para construir aplicações para um modelo de produto específico. O modelo do produto está relacionado com o número de pontos de comunicação em tempo real, que será permitido nesse projeto Veja "Versões e Licenças Action.NET" na pagina 217, para mais informações sobre o modelo do produto.

3. Pressione o ícone **Create New Project.**

Encontrando Projetos

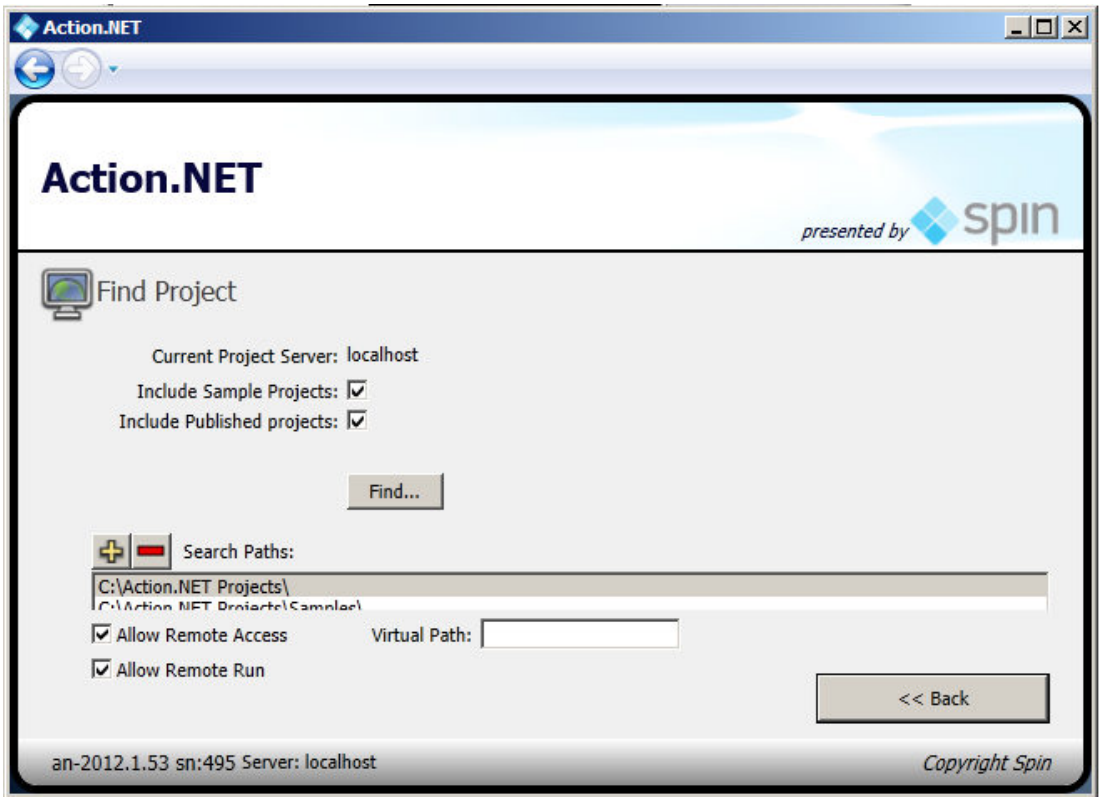
Quando se sabe que há outros projetos em outros diretórios nesta maquina ou em outros servidores na "NUVEM", poder-se utilizar esta ferramenta para a localização do projeto.

Ao clicar em **Find Projects** é mostrada a janela seguinte.

Nesta janela pode-se especificar em quais diretórios o Gerenciador de Projetos deve procurar por projetos Action.NET.

Ao acessar o computador local, o Action.NET procura por projetos na pasta padrão e em qualquer pasta onde você criou um projeto. Se você copiar um arquivo de projeto para uma pasta, Action.NET não conhece esta pasta como sendo de projetos. Você pode então adicionar essa pasta ao caminho de pesquisa, de modo que o Action.NET também procure projetos nesta pasta.

Você também pode remover as pastas do caminho de pesquisa, se você não precisa mais dessas pastas para projetos ou quer reduzir a lista de pastas.



Para encontrar projetos e gerenciar o acesso remoto:

1. Clique em **Find Projects**.
   - A janela exibe uma lista dos caminhos de pesquisa considerados.
2. Digite ou selecione as informações, conforme necessário.

Campo	Descrição
Current Project Server	Mostra onde o projeto atual esta sendo executado (read-only)
Include Sample Projects	Selecionado por “default”. Caminho com o diretório de projetos de exemplo. Desmarque se não quiser vê-los na lista
Include Published Projects	Selecionado por “default”. Desmarque se não desejar ver na lista os projetos já em estado “Published”.

Campo	Descrição
Find	Clique para navegar para um projeto. A pasta para o projeto selecionado é automaticamente adicionada à lista de caminhos de pesquisa e todos os projetos que estiverem nesta pasta serão apresentados na lista.
Search Paths	• Selecione a pasta cujas configurações de acesso você deseja alterar, em seguida, altere pressionando os sinais + ou - • Clique em + para adicionar uma pasta do caminho de pesquisa, para que todos os projetos desta pasta apareçam nesta lista. • Selecione uma pasta e clique em - para remover a pasta selecionada do caminho de pesquisa, de modo que os projetos desta pasta não serão mais exibidos na lista.

3. Clique **Back** para retornar à janela principal.
4. Esta página também contém as configurações para permitir que usuários em computadores remotos possam acessar os projetos, conforme descrito na próxima seção;

AVISO - Ao adicionar um projeto, ou uma pasta, todos os projetos da pasta selecionada são adicionados à lista de projetos na página principal.

Permitindo o acesso remoto

Os usuários remotos só podem ver e acessar projetos a partir do computador local que você configurar para acesso remoto. Se você deseja permitir o acesso remoto a um ou mais projetos, mas não para outros projetos, não se esqueça de colocar os projetos em pastas separadas. Quando você configura o acesso remoto (ou para a edição de projeto ou para a sua execução em tempo real), esta configuração inclui a pasta inteira. Todos os projetos em uma pasta configurada para permitir o acesso remoto será acessível a usuários remotos.

Para obter informações sobre como acessar um projeto remoto, consulte "Executando Projetos" na pagina 54.

A instalação do software inclui um servidor web (TWebServer) que permite o acesso remoto entre computadores com Action.NET instalados. Não é necessária qualquer outra configuração. Se você estiver usando o servidor Microsoft IIS, veja "Instalando serviços no IIS" na pagina 15.

Para encontrar projetos e gerenciar o acesso remoto:

1. Clique **Find Project.**
2. Selecione a pasta na lista de caminhos de pesquisa que deseja visualizar e modificar as permissões, selecione as informações, conforme necessário.

Campo	Descrição
Allow Remote Access	Selecionado por “default”.Desmarque para negar o acesso remoto para a pasta selecionada para a edição de projetos.
Allow Remote Run	Selecionado por “default”.Desmarque para negar o acesso remoto para a pasta selecionada para a execução em tempo real de projetos.
Virtual Path	Digite um titulo para fornecer acesso a apenas a pasta selecionada. Qualquer pessoa que acesse seu computador remotamente através do Action.NET pode ver todas as pastas que estão configurados para acesso remoto e que não têm um caminho virtual.

3. Clique **Back** para voltar a janela principal.

Accessando Projetos remotamente

Você pode acessar e editar projetos que estão configurados para acesso remoto. Os projetos podem estar em qualquer computador ou servidor que você possa acessar. Se você estiver habilitado para um serviço na nuvem, você pode acessar projetos configurados nele. A função de localizar o projeto não está disponível ao acessar um computador remoto.

Você também pode enviar um projeto para o computador remoto.

Para acessar remotamente um projeto:

1. Clique na aba **Server**.
   - Essa aba permite configurar e acessar projetos remotos.
2. Proceda de uma das seguintes formas:

Para acessara um projeto em um computador remoto	Para acessar um projeto sobre o serviço de nuvem
1. Selecione a opção Remoto. 2. Digite o endereço IP do computador em que o projeto está. 3. Se estiver configurado, digite o caminho virtual da pasta. • A pessoa que criou o projeto pode ter configurado o caminho virtual.	1. Selecione a opção de Cloud. 2. Digite sua conta e senha.

3. Clique **Connect**.
4. Se você deseja carregar um projeto, clique em **Upload Project** .
5. Clique na aba **Projects**.

- A lista de projetos disponíveis a partir do computador que você acessou será mostrada. Os projetos disponíveis devem estar em pastas que estão configuradas para acesso remoto.
  - Se você não digitou um caminho virtual, você vê todos os projetos em pastas configuradas para acesso remoto, mas não projetos em pastas que têm um caminho virtual configurado.
  - Se você digitou um caminho virtual, você vê todos os projetos das pastas configuradas com o caminho virtual que você digitou.

6. Clique duas vezes no projeto que você deseja abrir.

Executando Projetos

Você pode executar o projeto de qualquer uma das seguintes formas:

- Na janela principal Action.NET e lista de projeto, clique com o botão direito no projeto e selecione **Run Project**.
- Ao configurar o projeto, vá para **Run > Startup** e clique **Run Startup**.
  - Para mais informações, consulte "Executando a Aplicação" na pagina 195.
- Usando automaticamente as configurações de auto-inicialização na aba **Server**.

Atualizando projetos

Os Projetos retém informações sobre a versão do Action.NET que foi usado para sua criação. A menos que você atualizar o projeto para uma nova versão, o projeto abre sempre usando a versão que ele foi criado.

Quando você passa para uma versão mais recente do Action.NET, você pode atualizar um projeto para usar a versão mais recente.

Para atualizar um projeto:

1. Inicie o Action.NET:
   - Veja "Iniciando o Action.NET" na pagina 13.
   - A janela principal é mostrada. Quaisquer projetos existentes são exibidos na parte inferior da página.
2. Selecione o projeto que você deseja.
3. Para qualquer projeto que tenha um ProductVersion mais antigo do que sua versão atual, clique com o botão direito e selecione **Upgrade Project**.
   - Se esta opção não estiver disponível, o projeto já está na mesma versão que a versão atual do Action.NET.

- O sistema mantém o projeto original e adiciona o número da versão original para o nome do arquivo, assim você ainda pode abrir e executá-lo com a versão do produto com o qual o projeto foi desenvolvido. O sistema cria um novo projeto com o nome original do projeto, importando todas as configurações da versão mais recente do produto.

Configuração de projeto

Toda a configuração do projeto é executada em um aplicativo integrado, que fornece as facilidades de geração de tags, telas, configuração de canais de comunicação, e demais necessidades, de forma moderna, interativa e extremamente amigável.

Ferramentas de engenharia

Quando você clica em "**Open Project**" será mostrada uma janela com as ferramentas de configuração do projeto, ou como costumamos chamar, as ferramentas de engenharia.

A visão geral da interface do usuário de configuração do projeto está em "Área de Trabalho de Engenharia" na página 59.

Configuração orientada para SQL

Toda a configuração do projeto, telas com desenhos, imagens gráficas, scripts, definições de tags, o mapeamento de endereços em IEDs (dispositivos externos), e assim por diante, são salvas em um único banco de dados embutido orientado para SQL. Esse arquivo é criptografado para sua segurança e deve ser aberto e editado usando somente as ferramentas de engenharia de projeto.

A configuração do projeto é salva em um arquivo com extensão TPROJ. No âmbito do processo de **Publishing**, descrito em "Controle de Versão de Projetos" na página 199 , você também pode criar versões "read-only" do projeto, destinadas apenas para a execução em tempo real, que terão a extensão TRUN.

Multi-usuário, multi-Projeto

O sistema permite que muitos projetos possam ser abertos ao mesmo tempo. Se você já tem um projeto aberto, basta selecionar outro projeto na lista de Seleção de Projetos e abri-lo. Quando uma segunda instância do mesmo projeto é aberta, ela será aberto como somente leitura.

A fim de permitir usuários simultâneos você deve ter um Servidor de projetos licenciado para vários engenheiros. Na aba de licença, na caixa Grupo de Engenharia, você tem o número de usuários da ferramenta de Engenharia , que podem acessar simultaneamente o servidor selecionado ou computador local.

Informações sobre o Projeto

Cada arquivo de projeto tem incorporado metadados com as principais informações do projeto. Ao utilizar as ferramentas de engenharia, estas informações são mostradas na página Info-Project-Version.

As configurações do projeto que podem ser definidos pelo usuário, como a família e modelo do produto e as opções padrão pode ser definidas na página Info-Project-Settings na Ferramentas de Engenharia.

Para ver e alterar configurações do projeto, abra o Projeto com a ferramenta de configuração:

1. Vá para **Info> Version** .

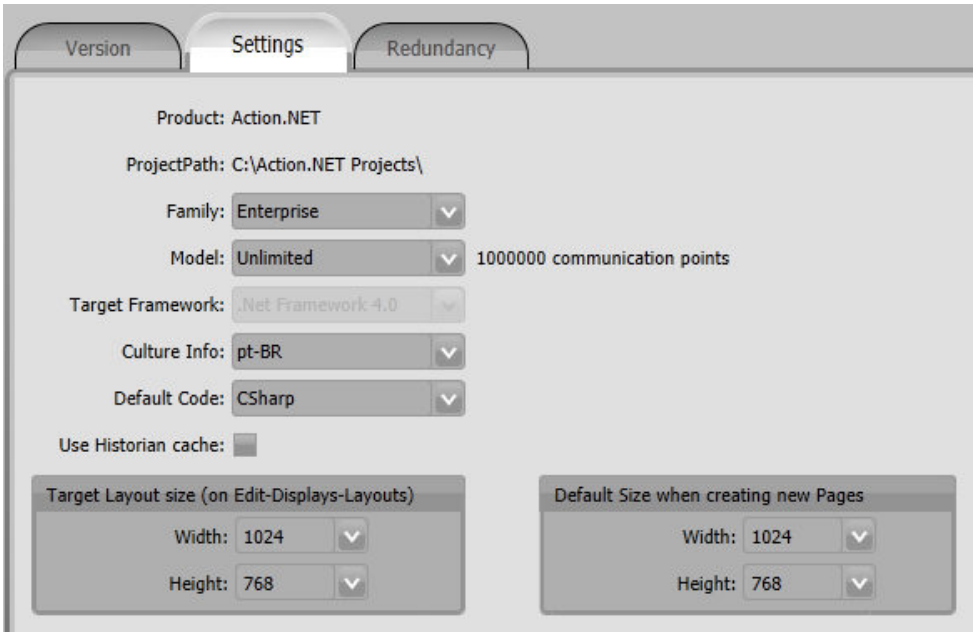
A maioria das informações é somente para leitura.

Configuração do Projeto (Settings)

Você pode visualizar e alterar as configurações do projeto que você definiu quando criou o projeto.

NOTA - A alteração de algumas configurações do projeto podem não ser possíveis porque levariam a condições não válidas. Por exemplo, alguns recursos só estão disponíveis na versão Enterprise.

Para ver e alterar configurações do projeto , abra o Projeto com as ferramentas de configuração e vá para **Info> Project> Settings.** É mostrada a página da figura a seguir.



Campo	Descrição
Product	Nome do produto utilizado para criar o projeto (Read-only)
ProjectPath	Caminho da pasta em que está o arquivo do projeto (Read-only)
Family	Veja "Versões e Licenças Action.NET" na pagina 217, para mais informação sobre as família do produto. Uma descrição básica para a família do produto é: • **Enterprise**—Funcionalidade completa do Action.NET • **HMI**—Use para aplicações menores e sistemas embarcados. • **Express**—Use somente para fins de teste ou demonstração. • **OPCServer**—Para criar um projeto com objetivo apenas de executar como um servidor OPC, fornecendo dados para outras aplicações a partir dos drivers de protocolo nativos. A alteração da família de um projeto, pode não ser possível porque poderia levar a condições não válidas. Por exemplo, alguns recursos só estão disponíveis na versão Enterprise.
Model	O modelo do produto está relacionado com o número de pontos de comunicação em tempo real, permitido nesse projeto. Veja "Versões e Licenças Action.NET" na pagina 217, para mais informações sobre o modelo do produto.
Culture Info	Selecione o idioma para a aplicação em tempo de execução. Esta configuração controla a formatação de datas, números e outras informações em conformidade com o idioma selecionado. Você pode alterar essa configuração mais tarde, se necessário.
Target Framework	Somente leitura. Mostra a versão do Microsoft Net Framework necessária para o projeto.
Default Code	Se o campo estiver habilitado, selecione a linguagem de programação (C# ou VB.Net) que você deseja usar para a criação de scripts personalizados. Você pode alterar essa configuração mais tarde, se necessário.
Use Historian cache	Selecione esta coluna se desejar que os dados gravados no historiador sejam em formato de "blob" binário, ao invés de valores em formato acessível. **AVISO -** Alterar essa configuração muda o formato de armazenamento e os dados no formato anterior não são convertidos automaticamente.
Target layout size	Selecione a resolução padrão para novos layouts a serem criados no projeto. Esta alteração não afeta as páginas já existentes.
Default size when creating new pages	Selecione a resolução padrão para novas telas a serem criadas no projeto. Esta alteração não afeta as páginas existentes.

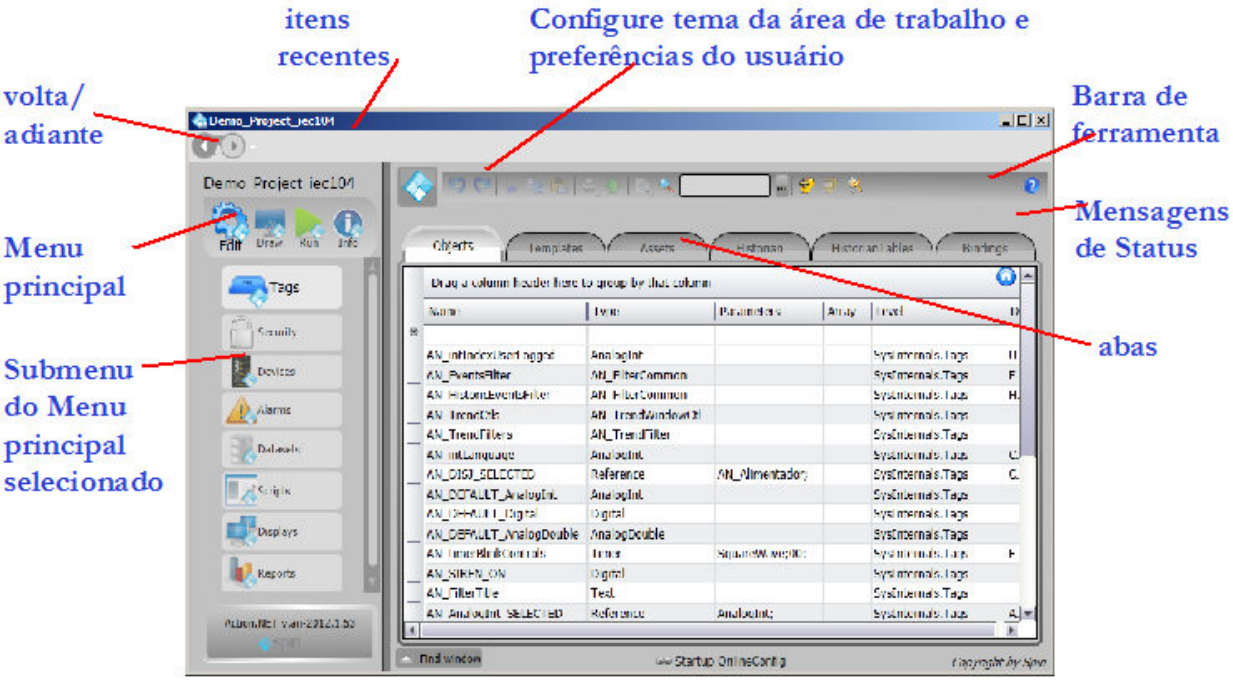
Área de Trabalho de Engenharia

As seguintes seções descrevem como criar e gerenciar projetos:

- "Visão geral da Área de Trabalho" na pagina 59
- "Navegando na Área de Trabalho" na pagina 62
- "Trabalhando com tabelas" na pagina 65
- "Usando a Referência Cruzada" na pagina 70
- "Intellisense e alteração de Nomes" na pagina 71
- "Dicionários e Localização" na pagina 71
- "Assistentes de Importação" na pagina 75.
- "Configurando Preferências" na pagina 76

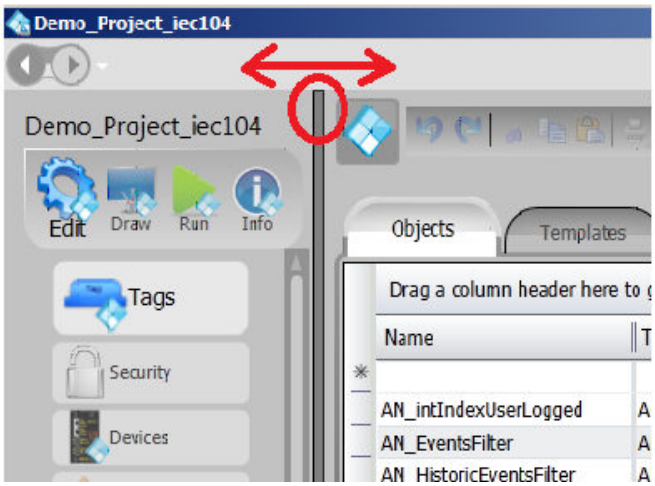
Visão geral da Área de Trabalho

A aplicação TManager.exe fornece acesso para editar a configuração de um projeto em Action.NET. A interface de usuário é semelhante a um navegador, com um botão para voltar, uma lista de itens recentes, e outras ferramentas de navegação. A Figura mostra as principais partes da página.



Grid Splitter

Você pode arrastar a barra vertical com o mouse para ajustar o tamanho do quadro de menu à esquerda, quando você quer ter mais espaço na área de trabalho.

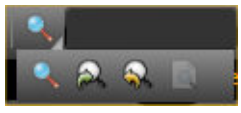


Usando a barra de ferramentas

A barra de ferramentas na parte superior de cada página tem todos os botões de edição como desfazer, refazer, recortar, copiar e colar, bem como botões para funções comuns que são exclusivas para Action.NET.

Além disso, alguns botões têm opções adicionais quando você clica com o botão direito neles. Por exemplo, os botões de **Imprimir** e **Procurar** tem opções adicionais, como mostrado na Figura.

Opções de botão direito na barra de ferramentas:



- As opções de **Imprimir** permitem imprimir ou salvar em um arquivo CSV.
- As opções de **Procurar** permitem que você encontrar, vá para o anterior, e ir para o próximo. Para mais informações, consulte "Trabalhando com tabelas" na pagina 65.

Desfazer e Refazer alterações

Os botões **Desfazer e Refazer** na barra de ferramentas permitem desfazer ou refazer qualquer operação. Ao editar tabelas, desfazer e refazer estão disponíveis para todas as mudanças durante a sua sessão de edição.

Ao editar documentos nas áreas de **Draw** (Desenho), **Reports** (Relatórios) e de **Scripts**, desfazer e refazer estão disponíveis até que você salve as alterações.

Cortar, Copiar e Colar inteligentes

Os botões Cortar, Copiar e Colar são sensíveis ao contexto: quando o foco está em um DataGrid a ferramenta vai cortar / copiar e colar as linhas selecionadas na tabela. Quando o foco está sobre a ferramenta de desenho, as funções serão executadas sobre os objetos selecionados.

A razão para a palavra inteligente é que você pode copiar e colar linhas de e para o Excel, além de outras tabelas no espaço de trabalho, mesmo que as colunas não sejam todas iguais. Ao copiar e colar, as colunas os cabeçalhos da mesmas também são copiados para o clipboard, então o sistema é capaz de colocar a informação na coluna correta, não importando a ordem que as colunas estejam sendo exibidas ou não., ou se não existem algumas colunas na tabela de destino.

Dica: Ao copiar de Excel certifique-se de selecionar os títulos de linhas. Você também pode copiar linhas a partir da tabela de **Tags** para a tabela de definição do **Historiador** diretamente para definir que deve ser feito o registro de dados dessas tags..

Imprimir, salvar em um arquivo ou Importar

Estes comandos permitem que você imprima o documento selecionado (um DataGrid ou um desenho). O botão de exportação ao ser pressionado, salva DataGrids para arquivos CSV e telas gráficas para arquivos JPG. O botão de importação irá importar arquivos CSV para o DataGrid.

Encontrar objetos

Você pode precisar encontrar onde você usou um objeto específico, como um **Tag**, em um projeto, o espaço de trabalho de Engenharia oferece três formas de pesquisa.Botão

Botões **Find String** e **Find Object**

- O botão **Find String** serve para a busca de uma cadeia de caracteres específica na aba atualmente selecionada da tabela DataGrid. Digite o texto que deseja pesquisar na caixa de texto e pressione o botão **FindString**.
- O botão **Find Object** na barra de ferramentas acessa o Objeto de Referência Cruzada, que permite localizar onde os objetos são usados no Projeto e vai diretamente para cada local. Para mais informações sobre referência cruzada, consulte "Usando a Referência Cruzada" na pagina 70.

Finalmente, você pode aplicar filtros dinâmicos na tabela selecionada, apenas digitando a máscara no campo de entrada. Exemplo: * abc * irá filtrar para mostrar apenas as linhas com as letras "abc" em qualquer posição do nome; ab * irá filtrar para mostrar as linhas começando com "ab". Clique no botão **Home** para limpar todos os filtros e redefinir todas as colunas do DataGrid para visualização no formato padrão do sistema.

Nova Tag e Propriedades de uma Tag

Nova Tag: permite criar uma nova tag no banco de dados em tempo real.

Propriedades da Tag: Mostra as propriedades da Tag Selecionada no TextBox.

Logon de segurança

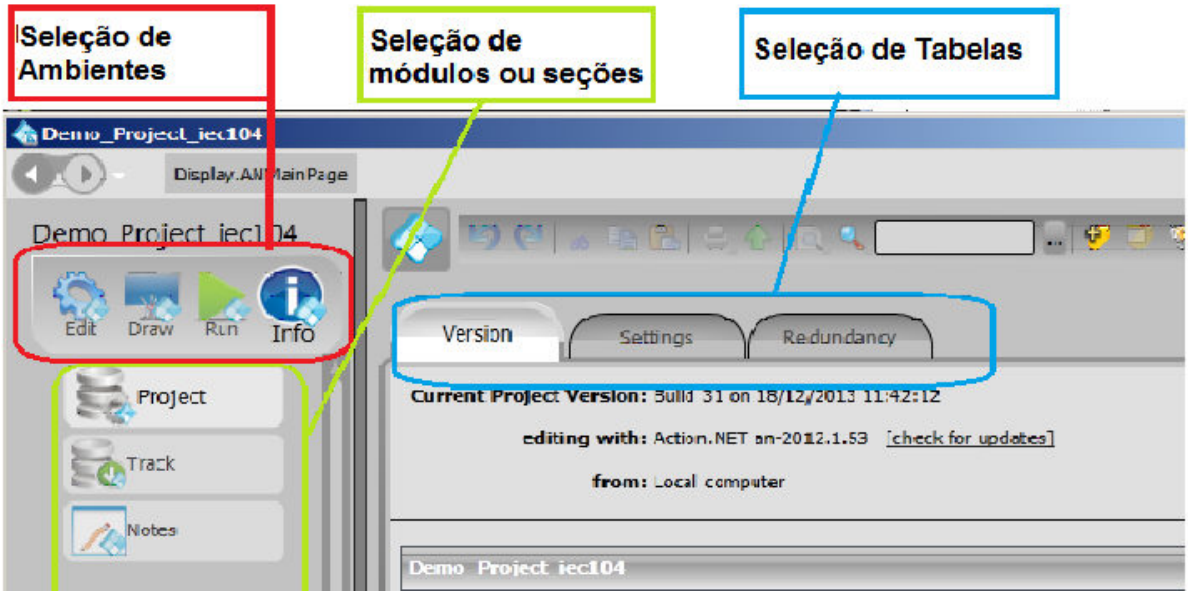
Ao passar o mouse sobre o ícone de Logon (chaveiro) é mostrado o nome de usuário conectado no momento. Se clicar no ícone abre-se a caixa de diálogo Logon. Por padrão, quando o sistema é iniciado, o usuário **Guest** é registrado. As permissões de cada usuário são definidas como descrito em "Segurança e Usuários" na pagina 97

Botão de Ajuda

O botão **Ajuda** navega no espaço de trabalho para a seção Info-Ajuda, que dá acesso ao arquivo de ajuda local, assim como permite fazer buscas no conteúdo online.

Navegando na Área de Trabalho

Você navega no Action.NET usando o ambiente de área de trabalho, seções e abas.



Qualquer interface de usuário do Espaço de Trabalho de Engenharia é, portanto, identificada por três nomes: o Ambiente (menu principal no canto superior esquerdo), a Seção (ou sub-menu) e a aba (ou a Tabela).

Para descrever como chegar a um local específico onde você quer executar funções, usamos uma notação abreviada.

Por exemplo, ao invés de dizer:

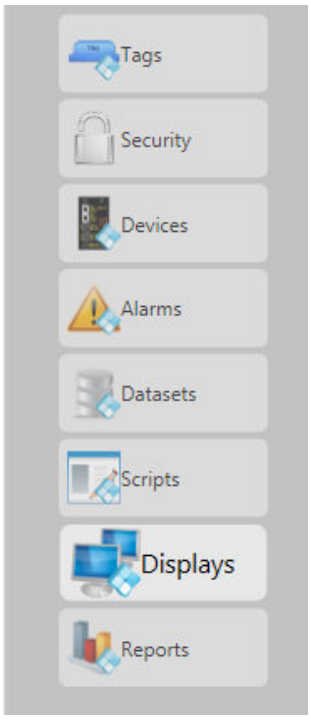
Clique em **Edit** , depois clique em **Tags**, em seguida, clique na aba de **Objects**.

Nós dizemos:

Vá para **Edit> Tags> Objects**.

Esta notação abreviada rapidamente informa como navegar para o menu principal correto, seções, e abas.

O botão EDIT no espaço de trabalho do módulo de Engenharia permite o acesso as tarefas de configuração dos módulos Action.NET



Para definir, verificar ou editar os "**Tags**" dos pontos a serem utilizados no projeto.

Security - Para definir as permissões e níveis de acesso dos usuários do projeto.

Devices - Para configurar protocolos de comunicação e IEDs (equipamentos de comunicação e de aquisição de dados) que serão utilizados pelo projeto

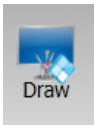
Para configurar itens de **Alarme** e Grupos de Alarmes para a geração de alarmes sob condições definidas.

Definição de fontes de dados como arquivos bancos de dados, tabelas específicas e "queries" pré-definidas.

Para configuração de **"Scripts"** que poderão ser em C-Sharp ou VB.Net, (em dot NET) compilados e ligados dinamicamente antes da execução dos projetos.

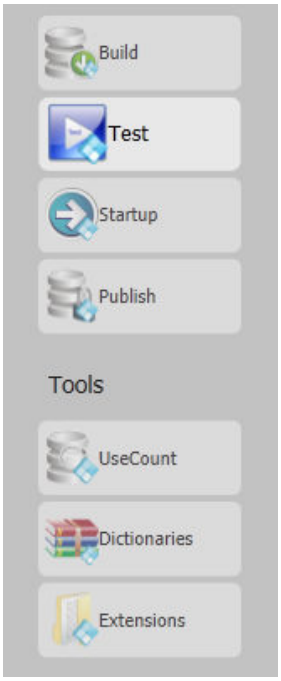
Para configurar ou selecionar telas gráficas dinâmicas, **Displays,** que vão mostrar em tempo real o estado dos processos supervisionados

Para configurar Relatórios, que podem incluir símbolos gráficos atualizados, em formato XPS, HTML, ASCII e Unicode.



Para configurar telas gráficas animadas, deve-se clicar no ícone **Draw** (*Desenhar)* no Menu principal. O símbolo aparece aumentado após o clique.

O ambiente de execução fornece acesso a todos os recursos de execução do projeto.



Build deve ser utilizado para solicitar a preparação do Projeto para execução em tempo real, e o salvamento da história do projeto, incluindo compilações e "builds" de scripts em bibliotecas de código executável.

Teste para testar o projeto e verificar sua execução em tempo real antes de publicá-lo para uso no Servidor real.

StartUp Permite especificar ferramentas de diagnose e os módulos a serem utilizados para testar aplicativos e scripts em execuções do projeto

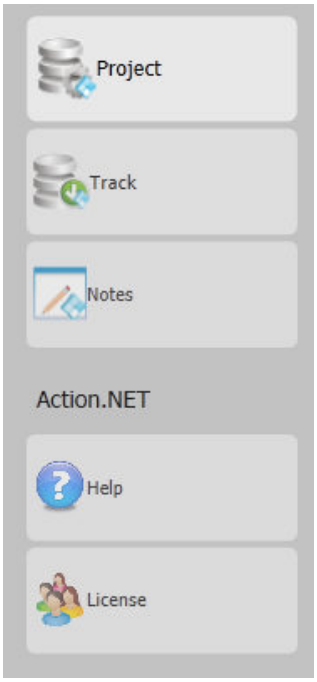
Publish - Publicar o projeto para obter o arquivo final de instalação no servidor. Obtém-se um novo identificador da Revisão Publicada do projeto.

Apresenta um sumário do projeto como numero de Tags utilizados. Fornece também um relatório de referência cruzada dos objetos do projeto.

Para a customização do idioma a ser utilizado nas mensagens de alarmes e textos de telas do projeto.

Importar dados de configuração já criados para Servidores OPC, de quaisquer tabelas de configuração em formato CSV, ou mesmo de um projeto completo. Configurar o uso de aplicativos externos (Modelos) compartilhando dados em tempo real.

O ambiente de Informações fornece acesso a detalhes de configuração atual do projeto



Mostra as versões do produto e do projeto, a localização do projeto, a linguagem utilizada e vários outros detalhes .

Para se obter informações sobre as Publicações do projeto, números de versões e história de alterações realizadas.

Para preenchimento de notas e observações importantes para outros usuários que estejam desenvolvendo o projeto.

Para chamar o Auxilio On-line de Referência do Action.NET

Mostra informações específicas do Licenciamento corrente do produto. Uma aba de Registro facilita o registro on-line do software.

Trabalhando com tabelas

Muitas abas no Action.NET exibem as informações em forma de Tabelas. Para tal utilizam um espaço denominado "DataGrid", que possui embutido várias funcionalidades que tornam muito amigável esta interface. Todas as tabelas permitem manipular, copiar, colar, e importar dados da mesma forma padronizada. As seções a seguir descrevem como usar estas funções:

- "Tabelas de Edição" na pagina 65
- "Classificação, filtragem e agrupamento" na pagina 66
- "Adicionando e removendo colunas" na pagina 68
- "Definição de colunas comuns" na pagina 68
- "Copiando e Colando Linhas" na pagina 69
- "Importando Dados" na pagina 69

Tabelas de Edição

Action.NET utiliza tabelas para a inclusão e atualização da maior parte de dados de configurações de um projeto no ambiente do menu **Edit**, como **tags**, **alarmes, usuários** . Na maioria das tabelas, você pode adicionar e editar as informações da mesma forma padronizada.

Para adicionar um novo item da tabela (linha):

1. Clique a primeira coluna da linha de inserção.
   - A linha de inserção é a primeira linha da tabela e tem um * ao lado dela.

   - Após clicar em uma coluna na linha, o * torna-se um ícone de lápis, indicando que agora você está editando essa linha.

2. Execute um dos procedimentos a seguir na coluna em que você clicou:
   - Se a coluna for um campo de texto, digite as informações.
   - Se a coluna for um campo de seleção, clique novamente, em seguida, selecione os itens da lista drop-down ou janela pop-up exibida.
3. Tecle **Tab** para movimentar o cursor entre as colunas da na linha.

Para editar uma linha da tabela existente:

1. Clique na coluna na linha que você deseja alterar.
2. Execute um dos procedimentos a seguir na coluna em que você clicou:

- Se a coluna for um campo de texto, digite as novas informações. O sistema substitui o texto que estava no campo. Para alterar parte do texto, clique novamente, e faça as alterações.
- Se a coluna for um campo de seleção, clique novamente, então selecione os itens da lista drop-down ou janela que exibe pop-up.

Mais de um usuário pode editar um projeto ao mesmo tempo. O sistema salva todas as alterações no servidor do projeto e propaga as mudanças para os outros usuários. Se dois usuários modificarem a mesma tabela de configuração ou documento, as alterações feitas pelo que salvar mais tarde vão substituir todas as alterações anteriores.

Dicas e Validação

Em todos as seções do Action.NET, os campos de dados e os botões têm embutidas "dicas" que contêm informações sobre a sua função. Algumas destas dicas incluem também um exemplo que pode ajudá-lo. O sistema também valida os dados que você entra na maioria dos campos, especialmente em tabelas. Se aparecer um contorno vermelho, ao redor da célula da tabela em que foi feita uma digitação, esta entrada não é válida.

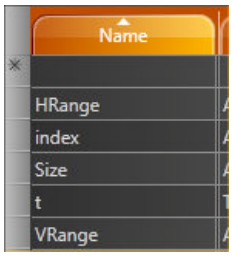
Classificação, filtragem e agrupamento

As informações existentes em muitas abas no Action.NET são exibidas em tabelas cujas linhas podem ser classificadas, filtradas e agrupadas.

Por default, os itens na tabela são listados na ordem em que você os criou.

Para ordenar uma tabela:

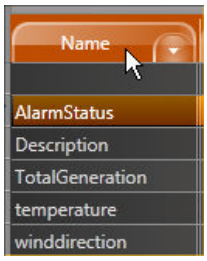
1. Em qualquer tabela, clique no cabeçalho da coluna que deseja usar para classificar.
   - Os itens agora estarão classificados em ordem alfanumérica crescente com base nessa coluna.



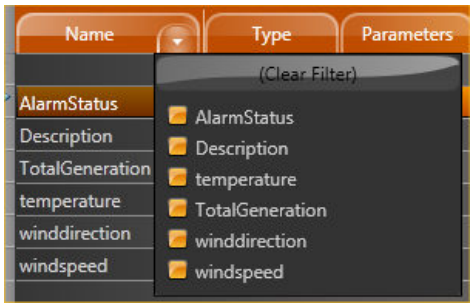
2. Clique no cabeçalho da coluna novamente para classificar em ordem alfanumérica decrescente. Uma seta sobre o titulo da coluna mostra se a ordem é ascendente, apoonta para cima, ou descendente apontando para baixo.
3. Clique no cabeçalho da coluna novamente para voltar ao modo padrão.

Para filtrar uma tabela:

1. Em qualquer tabela, passar o mouse sobre o cabeçalho da coluna que deseja usar para filtrar.
   - Uma seta para uma lista drop-down é exibida ao lado do título da coluna.



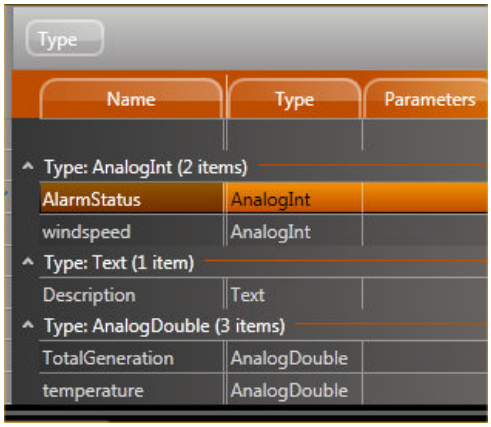
2. Clique na seta para ver a lista de itens na coluna.



3. Selecione os itens que você deseja incluir no filtro.
4. Para remover o filtro, clique na lista drop-down e clique em (Clear Filter) .

Para agrupar itens (linhas) em uma tabela:

1. Em qualquer tabela, arraste um cabeçalho de coluna para a área acima das colunas para agrupar por essa coluna.



2. Para remover o agrupamento, arraste o nome da coluna para a área de cabeçalhos de coluna.

Adicionando e removendo colunas

Tabelas em Action.NET exibem normalmente um conjunto padrão de colunas, mais geralmente utilizadas. Cada tabela pode ter colunas adicionais escondidas que você pode mostrar, ou você pode solicitar para ocultar.

Você também pode alterar a ordem das colunas. Para conhecer as definições de colunas que estão disponíveis em todas as tabelas, veja "Definição de colunas comuns" na pagina 68.

Para adicionar ou remover uma coluna:

1. Em de qualquer tabela, clique com o botão direito na área de cabeçalho da coluna.
   - Uma lista de todas as colunas disponíveis para essa tabela é exibida. Colunas com uma marca de seleção ao lado deles exibem atualmente na tabela.
2. Marque ou desmarque as colunas na lista.
3. Clique fora da lista para a fechar.
4. Para alterar a ordem das colunas, arraste um cabeçalho de coluna para um novo local.
5. Para voltar para a Tela padrão, clique em

   ■Restaurar as configurações padrão de grade.

Definição de colunas comuns

Várias colunas comuns estão disponíveis em muitas Tabelas Estas colunas são definidas abaixo, em vez de repetir-los na descrição de cada tabela em itens seguintes.

Table 1: Descrições das colunas comuns

Coluna	Descrição
ID	ID do registro de banco de dados atribuído a este objeto na tabela. Os números são atribuídos iniciando de 1 para o primeiro objecto criado e continuam crescendo sequencialmente conforme necessário. Os IDs são exclusivos apenas para uma tabela específica..
Level	Selecione o nível que você deseja associar a um Tag na aba Níveis (Levels ou Assets). Veja "Criando Níveis (Assets)" na pagina 88.
Category	Selecione uma ou mais categorias para este atributo. Veja "Dicionário de Categorias" na pagina 74.
DateModified	Data em que este objeto foi modificado pela ultima vez
DateCreated	Data em que este objeto foi criado inicialmente.
Description	Para se digitar uma descrição deste objeto (comentário opcional)

Copiando e Colando Linhas

Você pode copiar e colar linhas de e para tabelas em Action.NET. As funções copiar / colar permitem copiar linhas de uma tabela para outra dentro do Espaço de Trabalho de Engenharia. Por exemplo, você pode copiar os **Tags** e cola-los na aba Historiador ou na aba Itens em Alarmes.

Você também pode copiar e colar de e para uma planilha.

Para copiar linhas para outra aba:

1. Em qualquer tabela, selecione uma ou mais linhas.
2. Pressione o botão direito do mouse e selecione Copiar linhas selecionadas .
3. Vá para a outra aba destinatária da copia.
4. Clique o botão direito do mouse e selecione Colar linhas.

Para copiar linhas de uma folha de cálculo:

1. Em qualquer tabela, selecione uma ou mais linhas.
2. Clique o botão direito do mouse e selecione Copiar linhas selecionadas.
   - A cópia inclui automaticamente todos os títulos de coluna disponíveis para a aba em que você está.
3. Vá para uma planilha e selecione Colar linhas.

Para colar as linhas de uma folha de cálculo:

1. Configure as colunas da planilha, seguindo os passos de cópia acima.
2. Digite as informações que deseja.
3. Copie as informações, incluindo os títulos, a partir da planilha.
4. Sobre a tabela correspondente destinatária, clique com o botão direito e selecione Colar linhas.
   - A operação de colar, coloca automaticamente os dados nas colunas correspondentes. Todas as células vazias são ignoradas.

Importando Dados

Você pode importar um arquivo do tipo .CSV (valores separados por vírgulas) em qualquer tabela. Os títulos das colunas da planilha devem corresponder aos da tabela para a qual você está importando a planilha. A melhor maneira de garantir a correspondência de nomes de colunas é copiar algumas linhas da tabela em uma planilha em primeiro lugar, em seguida, editar a planilha inclusive incluindo novas linhas que forem necessárias e depois colar a planilha sobre a tabela Para obter informações sobre como copiar linhas, consulte “Copiando e Colando Linhas” na pagina 69.

Para importar os dados:

1. Crie os dados em uma planilha.

- Se você estiver criando **Tags** ou outros objetos, você pode nomear os objetos, conforme necessário. Se você está preenchendo outras colunas, os dados nas colunas devem corresponder ao que o Action.NET espera.

2. Salve o arquivo como um arquivo..CSV.
3. Da tabela correspondente no Action.NET, clique em **import from a CSV file**.
4. Navegue até o arquivo CSV. e selecioná-lo.
5. Clique em **Open** .
   - O sistema importará os dados na tabela.

Usando a Referência Cruzada

Você pode obter a informação de quantas vezes cada objeto do projeto é usado, bem como obter uma lista de objetos que não são usados. Também, a partir de uma pesquisa pelo nome do objeto você consegue chegar ao documento ou tabela onde ele está sendo usado.

Para encontrar os objetos:

1. Siga uma das seguintes opções:
   - A partir da tabela que tem uma ocorrência do objeto que você deseja encontrar, selecione o objeto. Por exemplo, selecione o **Tag** que você deseja encontrar em outras partes do projeto.
   - O campo **Find** na barra de ferramentas será automaticamente preenchido com o texto do **Tag** selecionado.

   - No campo **Find** barra de ferramentas, digite o tipo de objeto (tag), seguido por um. (ponto), seguido pelo nome do objeto (o nome do Tag, neste caso).
   - À medida que você digita no campo, o sistema seleciona e mostra à primeira entrada de objetos no sistema. Após o primeiro, lista todos os objetos desse tipo.
   - Ao lado do campo **Find** clique no botão mostrado a baixo. A Janela **Selecionar objeto** mostra uma lista de todos os tipos de objetos.

   - Quando você seleciona um tipo de objeto, aparece uma lista com todos os nomes de objetos existentes do mesmo tipo..
2. Clique **Find** ou a partir da janela Selecionar objeto, clique em **OK**.
   - Os resultados da pesquisa serão mostrados em uma lista na parte de baixo da página.
3. Um duplo-Clique sobre qualquer linha na lista de resultados fara aparecer este item.
4. Para fechar o painel de resultados, clique em **Find Window** na parte inferior esquerda da página.

Para monitorar o uso do objeto:

1. Vá para **Run > Tools > UseCount** .

Para visualizar os objetos que não são usados, clique na aba **UnusedObjects.**

Intellisense e alteração de Nomes

Em todos os campos de entrada, seja nos DataGrids ou nas ferramentas de desenho, onde você pode usar nomes de Tags ou nomes de objetos, está disponível a facilidade que se denomina IntelliSense: Na entrada de dados sugere nomes válidos de acordo com as letras digitadas

O recurso Intellisense apresenta uma lista de objetos válidos de acordo com as letras que você está digitando e as entradas válidas para esse campo de entrada específico. Você pode inserir o item selecionado na lista pressionando "enter", espaço, a tecla ponto ou outras teclas de símbolos.

Os campos IntelliSense também tem dicas e validação nele embutidos. Colocando o mouse sobre um campo serão apresentadas as entradas esperadas e possíveis para esse campo. Quando você tem uma entrada inválida, o contorno do fronteira do campo fica em cor vermelha e a dica dá informação sobre o erro existente.

Denomina-se Refatoração o processo de alterar a definição de nomes de Tags, nomes ou identificadores de quaisquer outros objetos (como grupos de alarme, nós de dispositivo e de qualquer outro), depois que o nome simbólico já está em uso na aplicação.

Graças à configuração centralizada SQL, o Action.NET permite renomear qualquer objeto, bastando para tal, digitar o novo nome na tabela de configuração onde o objeto está definido. Automaticamente todas as referências a esse objeto serão alteradas, usando o novo nome, sem mais ações necessárias.

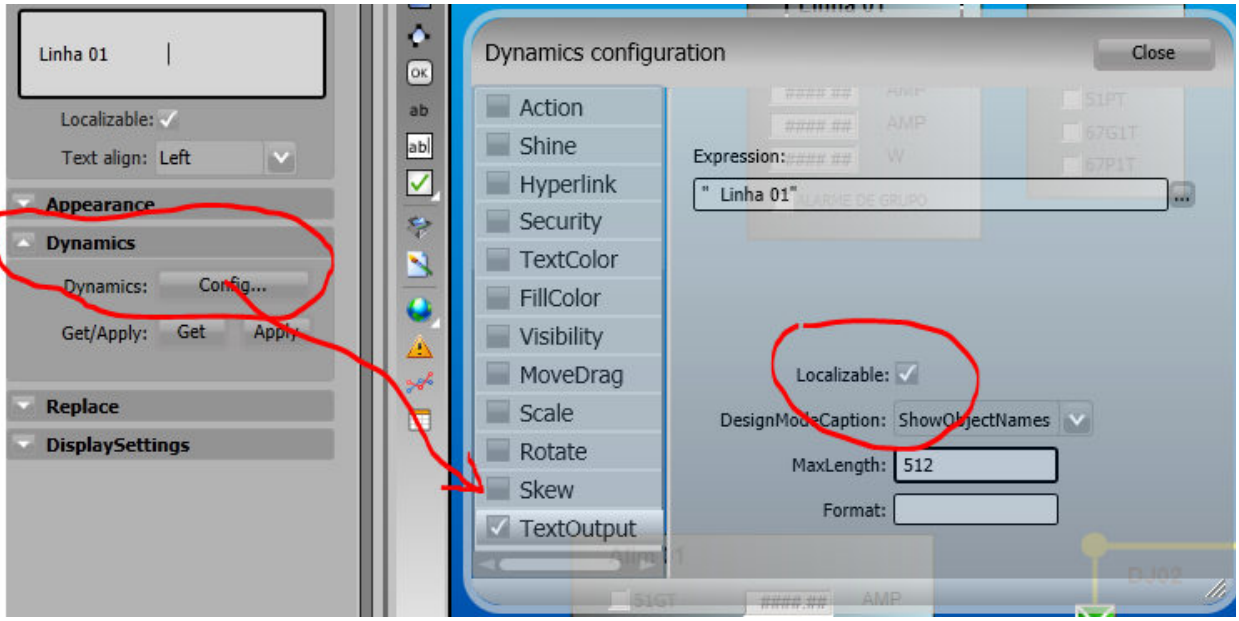
Por exemplo, para renomear um Tag que já está sendo usado em todo o projeto, apenas vá em **Edit-Tags-Objetos** e troque o nome na linha em que está definido. O mesmo se aplica para **Edit-Alarms-Groups**, os membros de Templates ou qualquer outro **Name** de objeto no sistema.

Dicionários e Localização

Localização da linguagem no Runtime

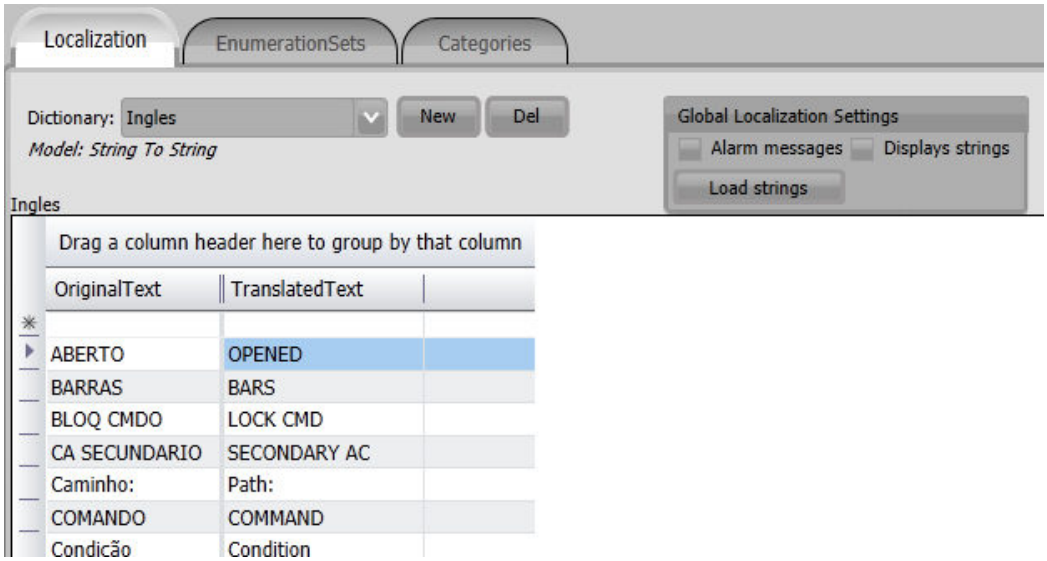
Neste contexto Localizar é transformar alguma coisa nos costumes locais , como a tradução de textos para o idioma local. Você pode mostrar o texto em seu aplicativo em tantas línguas necessitar. Basta você criar um dicionário para cada idioma.

NOTA - Para mostrar em tempo de execução texto "localizado" na interface do usuário, não se esqueça de selecionar a opção **Localizable** (janela **Dynamics Configuration**, opções TextOutput) para cada objeto de visualização das Telas que você precisa para localizar. Veja figura a seguir.



Para configurar a localização:

1. Ir para **Run > Dictionaries -> Localization**.
2. Clique **New**.
   - Aparecerá a janela Criar Novo Dicionário Localização.
3. Digite um nome para o dicionário.
4. Clique em OK.



5. Clique **Load Strings** para preencher a tabela com todos os controles que têm textos que foram definidos como **localizáveis**. Na coluna **TranslatedText**, digite o texto no idioma do dicionário. Marque **Alarm Messages** se também desejar que as mensagens de alarme sejam localizadas.

6. Crie controles de visualização em uma Tela ou escreva um script para usar o objeto chamado Client.Dictionary para configurar o dicionário ou opções de dicionário para o usuário. Quando o dicionário é trocado as correspondências de textos são alteradas em toda a aplicação.

As mensagens de alarme podem também ser incluídas na localização, se o check-box correspondente na definição da mensagem estiver marcada (ver figura acima).

Em **Info-Project-Settings**, você também pode definir qual o default para **Culture Information** que o aplicativo deverá usar. Há disponível para scripts a propriedade runtime `Client.CultureInfo` que permite você alterar esta definição em tempo de execução.

Alguns objetos de visualização podem obter a informação `Culture` diretamente do sistema operacional, por isso, o procedimento correto é ter tanto a configuração de sistema operacional e as configurações do projeto com a cultura alvo para executar a aplicação.

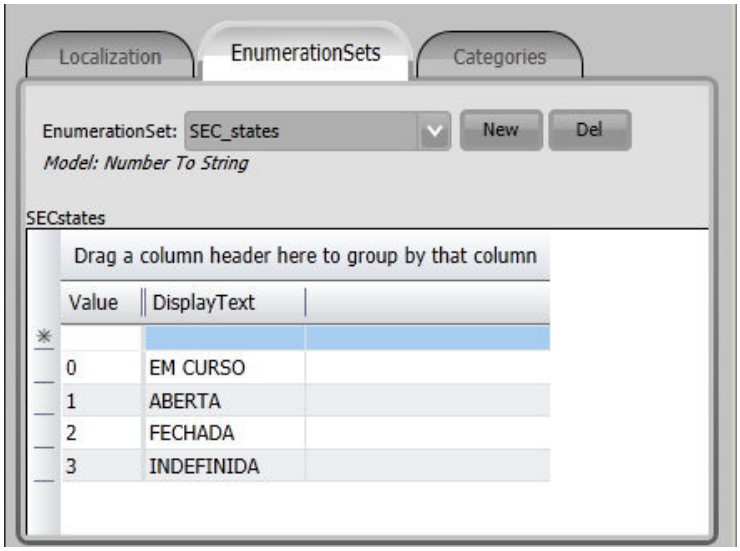
Conjuntos de Enumeração

Você também pode usar os dicionários para traduzir os valores dos tags em texto. Por exemplo, um dispositivo pode ter um valor de 0 (desligado) ou 1 (ligado). Ao invés de exibir "0" ou "1", você pode exibir as palavras "Ligado" ou "Desligado".

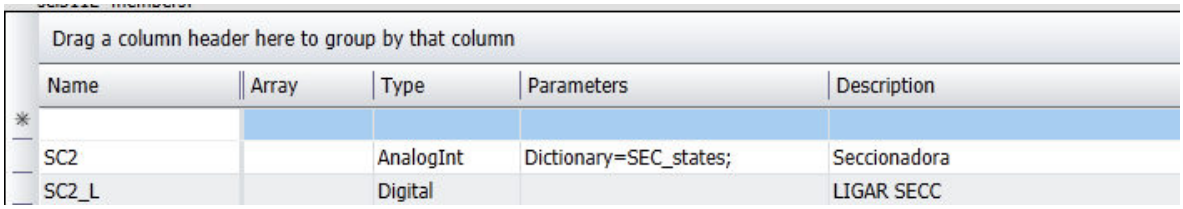
Para configurar conjuntos de enumeração:

1. Vá para **Run > Dictionaries > Enumeration Sets**.
2. Clique **New**.
   - Aparecerá a janela Criar Novo Dicionário.
3. Digite um nome para o dicionário.
4. Clique em **OK**.
5. Na coluna Valor, digite o valor numérico, e na coluna Texto traduzido, digite o texto a ser exibido.
6. Para os Tags que usam os valores numéricos com o Conjunto de Enumeração, configurar a coluna Parâmetros para usar este conjunto.
7. Ao configurar um componente de uma tela para usar estas tags, deve-se definir o TextOutput para um dos seguintes atributos:
   - Com tagName ou Tagname.Value para mostrar o valor numérico da tag.
   - Com Tagname.ValueAsString ou Tagname.DisplayValue para mostrar o texto do conjunto de enumeração.

Veja na figura um exemplo para a criação do Conjunto de Enumeração “SEC_States” para a definição dos estados de uma chave seccionadora:



Na definição do Tag veja a indicação para uso do dicionário do Conjunto de Enumeração:

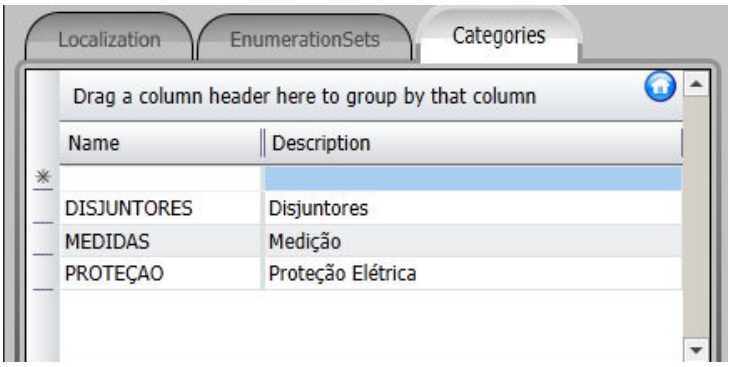


Dicionário de Categorias

Um tag pode pertencer a uma categoria pré-definida de objetos. Por exemplo: tags de trafos, tags de alimentadores, tags de proteção, podem ser categorias. As categorias devem ser pré-definidas no Dicionário de Categorias.

Você pode criar Categorias para permitir a classificação vários tipos de objetos da Engenharia. A maioria das tabelas de configuração, como Tags, Nós de Dispositivos, Grupos de Alarme, têm a coluna Categoria, onde você pode aplicar uma ou mais categorias para esse objeto.

1. Vá para **Run > Dictionaries > Categories**.
2. Edite as categorias desejadas para este projeto.



Um mesmo tag ou objeto pode pertencer a várias categorias.

Por exemplo: suponha que foram definidas duas categorias: TAGS_DE_TRAFOS e TAGS_DE_MEDIDAS. Poderíamos querer pesquisar, dentro de um código de script, os tags que são de pontos de medição de tensão nos trafos. No cadastramento destes pontos especificaríamos no campo **Categoria** destes tags estas duas categorias.

Uma vez a Categoria criada, passará a estar disponível na lista de existentes na coluna Categoria da aba Objeto, para a criação de Tags. Para cada tag escolha todas em que o mesmo se enquadra.

As Categorias de objeto também pode ser acessadas durante o tempo de execução para a filtragem e outras aplicações personalizadas, utilizando a propriedade Categoria dos objetos, como critérios para a escolha de objetos. Uma lista de todas as categorias definidas no projeto está disponível em tempo de execução na propriedade `Server.Categories`.

Assistentes de Importação

Na seção **Run-Extensions-Import** você encontra um conjunto de assistentes para importar componentes de projetos.

Este conteúdo exibido nessa interface são criados dinamicamente de acordo com a DLL encontrados na pasta IMPORT da instalação do projeto.

Integradores de Sistemas e outras empresas parceiras podem criar assistentes de importação, utilizando os Toolkits apropriados.

As ferramentas de importação já embutidas instalados com o produto são:

Importar projetos Action.NET

Adicione os componentes do projeto selecionado para o projeto atual.

Importar Rockwell ControlLogix

Cria os Tag Names, Tag Templates e faz o mapeamento dos endereços PLC, tudo de acordo com as informações disponíveis no arquivo L5K.

Importar configuração de um Servidor OPC

Cria os Tag Names, Tag Templates e faz o mapeamento dos endereços cliente OPC, tudo de acordo com as informações disponíveis, navegando para o servidor OPC

Importar de um sistema OSIsoft PI

Ao se conectar com servidores Historiador PI System, ele cria automaticamente o link com as PI Tag, elementos e atributos de servidores PI AF.

Quando conectando com PI AF, traz automaticamente a definição de todos os modelos e elementos ativos no banco de dados selecionado, assim como o mapeamento para comunicar com o servidor.

NOTA - O assistente de importação PI só é mostrado em instalações licenciados com o módulo de conexão PI habilitado

Importando Tag Templates em XML

Cria os Tag Names, Tag Templates de acordo com a estrutura de dados do arquivo XML, ele também cria a configuração Edit-Datasets-Files para ler esse arquivo

Importando classes dot NET de uma DLL

Cria Tag Templates que correspondem as classes dot NET selecionadas.

Importando projetos ActionView

Este assistente de importação pode ser utilizado para a importação de Tags, Alarm Items, Historian tables a tabelas de Points de um projeto ActionView.

Geração automatizada de projetos Action.Net

Este assistente de importação pode ser utilizado para a geração automática de Tags, Alarm Items, Historian tables a tabelas de Points. Para tal é necessário que se crie anteriormente Templates, Canais, Nodes e as necessárias Historian Tables. Também devem ser criadas Categorias indicativas de como devem ser usados os templates e Points. O assistente ao ser executados faz a "explosão", i. é cria as tabelas mencionadas. Para mais detalhes sobre este assistente veja o capitulo "Deploy - Geração automatizada de projetos" na página 283.

Configurando Preferências

Você pode configurar o tema de aparência do Action.NET e outras preferências.

Para configurar as preferências:

1. Clique no botão do aplicativo (o logotipo da Spin no topo da barra de ferramentas).

2. Altere ou selecione as opções desejadas.

Configuração	Descrição
Theme	Selecione o esquema de cores que você prefere no ambiente de engenharia. O tema também é útil se você abrir o mesmo projeto duas vezes (uma para editar e outro para ver as informações). Você pode aplicar um tema diferente para cada janela para distingui-los
Grid Fonts	Clique para ver as fontes disponíveis para a interface do usuário. Selecione o tipo de letra (fonte), o estilo, tamanho e, em seguida, clique em **OK**.

Configuração	Descrição
Auto-save document	Selecione a frequência com que você deseja que o sistema salve automaticamente as alterações ao editar scripts, usar as ferramentas de desenho e de criação de relatórios **Nota**: O sistema salva a maioria das mudanças automaticamente, independentemente desta definição. Essa configuração só é aplicável para as funções listadas acima.
Confirm on Build commands	Selecione para exibir uma janela de confirmação toda vez que você executar um BUILD no projeto.
Allow to duplicate Tags on	• Tabelas do Historiador -Selecione para permitir o mesmo nome de Tag mais de uma vez nas tabelas de historiador. Você pode querer Tags duplicadas com taxas diferentes ou em diferentes grupos. • Device Points-Selecione para permitir o mesmo nome de Tag ser utilizado mais de uma vez na tabela de POINTS. Você pode querer tag duplicados para diferentes PLCs ou diferentes taxas de amostragem, ou um mesmo Tag ser alimentado por um protocolo mestre e ser servido para outro sistema por um protocolo escravo (servidor)
Help file location	• Caminho de Instalação Local - Selecione para usar o arquivo de ajuda que foi instalado em seu computador com o software Action.NET • URL-Selecione se você quiser acessar a ajuda a partir do site, ou você copiou o arquivo de ajuda (. chm) para um servidor web e quer acessar a ajuda a partir do servidor web. Digite o URL para o arquivo no servidor web

3. Clique em **CLOSE** no canto superior direito.

Tags, Níveis e Tipos

Na interface de Engenharia do Action.NET e ao longo deste manual, o termo **Tag** refere-se a qualquer variável na base de dados, com seus atributos em tempo real ou com seus dados históricos associados. Os Tags geralmente são mapeados em registros ou pontos em IEDs, ou PLCs ou outro equipamento físico de um processo de produção. Um Tag também pode ser conectado a entradas em bancos de dados SQL, fontes de dados externos, ou um valor calculado internamente.

As seções a seguir descrevem como configurar tags:

- "Criando e Editando Tags" na pagina 79.
- "Criando Tipos Customizados (Tag Templates)" na pagina 85.
- "Criando Níveis (Assets)" na pagina 88.
- "Criando Categorias" na pagina 89.

Criando e Editando Tags

Tags são as variáveis do processo para a sua aplicação. Use os tags e suas propriedades para criar o modelo de dados para o seu processo. Você pode estender os tipos de tags disponíveis e criar novos tipos para atender às suas necessidades de aplicação, tais como dados de máquinas, estado dos equipamentos, medições de grandezas elétricas, ou a representação de todos os atributos de conjuntos hierárquicos ou ativos (Níveis ou Assets) em sua planta.

Para criar e editar as tags:

1. Vá para **Edit > Tags > Objects** .
   - Você também pode criar um Tag em qualquer lugar no espaço de trabalho do Action.NET clicando **New Tag** na barra de ferramentas.

DICA - Como qualquer outra tabela de configuração, você pode também importar arquivos CSV ou conteúdos copiar / colar diretamente de uma planilha do Excel ou outras aplicações. Verifique se o capítulo "Visão geral da Área de Trabalho" na pagina 59 para obter informações sobre a funcionalidade DataGrid para edição de tabelas.

2. Digite ou selecione as informações, conforme necessário, na linha de inserção da tabela.

Coluna	Descrição
Name	Digite um nome para o Tag. O sistema avisará se o nome não é válido mostrando o contorno do campo em cor vermelha. Nota: Se você editar o nome de um Tag existente, o sistema atualizará automaticamente todas as referências a este nome nos existentes em todo o projeto.
Type	Selecione um dos "Tipos de Tags pré-definidos" na pagina 82 ou utilize um tipo definido pelo usuário (Template). Veja em "Criando Tipos Customizados (Tag Templates)" na pagina 85.
Parameters	Se estiver disponível, configure os parâmetros. Os parâmetros variam com base no tipo do Tag.
Array	Quando este campo não está definido (em branco) o Tag é único. Não é uma matriz. Ao definir uma matriz com valor N será criada uma matriz de Tags com as posições de 0 a N. Por exemplo, ao criar uma matriz de Tags com o numero "5", a matriz é criada a partir de Tag [0] até o Tag [5], o que significa que 6 elementos são criados.
Description	Digite uma descrição para esta tag.

3. Para adicionar ou remover uma coluna, clique com o botão direito na área de cabeçalho da coluna e marque ou desmarque as colunas que serão visíveis.

Coluna	Descrição
EngUnits	Entre a unidades de engenharia da grandeza de uma medida que você deseja usar para essa tag.
StartValue	Insira um valor inicial para esta tag. Este é o valor com o qual o tag será inicializado quando for feita a primeira execução do sistema em tempo real.
Format	Digite um formato padrão para exibir os dados. Por exemplo: • N0-Número, sem casas decimais. • N3-número com três casas decimais. • X-hexadecimal (suportado apenas para tipos inteiros). • C-moeda. • Quando configurando a forma de Tela de dados nas telas, é possível definir o formato para cada campo individualmente, mas na maioria das situações, é mais fácil anexar a formatação padrão para a Tag diretamente Para mais informações, consulte "Formatos de Tags" na pagina 84.

Coluna	Descrição
Retentive	Selecione uma opção para salvamento do valor do Tag e suas propriedades internas ao banco de dados cada vez que o valor muda. Isso mantém o valor quando a aplicação é encerrada e faz com que o valor fique guardado e seja utilizado quando o aplicativo for novamente iniciado. Escolha uma das opções: • Nenhum-não retém o valor ou propriedades. • ValueOnly-Mantém apenas o valor. • Properties-Mantém todas as propriedades, incluindo o valor. • Properties Only - Mantém todas as propriedades, mas não o valor
Min	Digite o menor valor que é valido para este objeto
Max	Digite o maior valor que é valido para este objeto
Visibility	Selecione um dos modos de visibilidade de dados através do servidor OPC existente no Action.NET., para os projetos remotos • Private—O Tag é visível somente para o projeto local e se par redundante. • Protected—O tag é read-only sendo visível no Servidor OPC para projetos remotos e clientes OPC. • Public—O Tag é visível no Servidor OPC para projetos remotos e clientes OPC.
Domain	Define o domínio do valor do Tag para todo o projeto ou específico para cada cliente de visualização.. • Server—O valor do Tag value é o mesmo através de todo o projeto e todos os clientes de visualização remotos. A maioria dos Tags em um projeto devem ser deste tipo. • Client—O valor do Tag é local para cada computador executando um cliente de visualização remoto ou cliente web. • Pode se usar Tags locais para conter dados temporários específicos para um cliente individualmente. A utilização mais comum para tags locais é quando dados temporários são necessários para o gerenciamento de telas na interface do usuário. Os tags locais poderão ter valores diferentes em cada computador cliente.
Comment	Digite aqui opcionalmente qualquer comentário sobre este tag
[Outras colunas]	Para a definição de outras colunas que estão disponíveis na maioria das tabelas veja "Definição de colunas comuns" na pagina 68.

4. Continue adicionando quantos Tags quantos necessitar
5. No Action.NET você pode alterar o nome de um tag a qualquer momento, deste modo, é muto fácil criar um novo Tag clicando com o mouse na coluna **Name** da linha de inserção e teclar repetidamente Espaço e Enter. O sistema criará um novo Tag do mesmo tipo do ultimo criado. Você pode também usar o mouse na linha de inserção para selecionar o Tipo, pode clicar com o mouse no cabeçalho de qualquer outra coluna que também será criado um tag com um nome por default formado pelo tipo e seguido de um numero sequencial.

Tipos de Tags pré-definidos

A tabela a seguir apresenta os tipos de dados padrão, no Action.NET, seu correspondente em dot Net, e o intervalo de valores que cada tipo é capaz de assumir.

Tipode dados padrão

Tipo deTag	Tipo em .Net	Intervalo de valores
Digital	System.Int32	0 até 1
Integer	System.Int32	2,147,483,648 até 2,147,483,647
Long	System.Int64	2,147,483,648 até 2,147,483,647
Double	System.Double	-1.79769313486231570E+308 até -4.94065645841246544E-324 para valores negativos; 4.94065645841246544E-324 até 1.79769313486231570E+308 para valores positivos
Decimal	System.Decimal	0 até +/-79,228,162,514,264,337,593,543,950,335 sem ponto decimal ; 0 até +/-7.92281625142643375935439503 35 com 28 casas para a direita do ponto decimal; Menor não zero é +/-0.000
Text	System.String	0 to aproximadamente 2 bilhoes de caracteres Unicode
Timer	System.Int32	Mesmo intervalo dos Inteiros, mas com parametros embutidos para manipular temporizadores
DateTime	System.DateTimeOffset	de 12:00:00 midnight, January 1, 0001 to 11:59:59 P.M., December 31, 9999
TimeSpan	TimeSpan	Data Interval, em Days, Hours, Minutes, Seconds and Milliseconds, onde cada uma das propriedades pode conter um valor Double
Guid	Guid	GUID Identificador padrão unico
DataTable	System.Data.DataTable	Mantém uma DataTable na memória
Image	System.Byte[]	Pode guardar um arquivo de Imagem ou um arquivo de conteúdo binário. O tamanho máximo do vetor é o de um numero Long

NOTA - O tipo Decimal foi criado em. NET para permitir o cálculo com mais precisão do que o tipo Double, 24 de quatro dígitos contra 14, o que é necessário quando se lida com números altos e nenhum arredondamento permitido, como o cálculo de dinheiro e algumas medidas de engenharia que precisam ser muito grandes e muito precisas, porém as operações matemáticas em um Decimal pode ser 40 vezes mais lentas, do que um Double. Então, use o tipo Decimal apenas nos casos em que a precisão dupla não é suficiente.

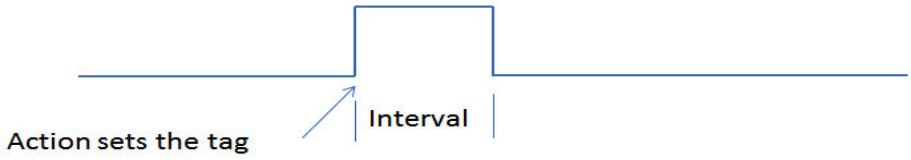
Temporizador (Timer)

É um valor inteiro (Built-In) que é gerado automaticamente pelo sistema para implementar vários tipos do temporizador:

- **SquareWave**: o valor alterna entre 0 e 1.

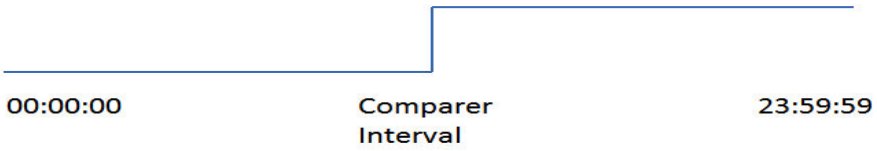
- **Pulse**: O tag muda para 0, em seguida, muda imediatamente a 1.

- **DelayOff**: O Tag se comporta como um PLC Timer Off. Durante o tempo de execução,



se você definir o tag com algum valor diferente do startValue, o Tag vai segurar este valor pelo período de tempo especificado no parâmetro Intervalo. Então o Tag volta ao seu startValue após o período de tempo (Intervalo)

- **Comparer**: O tag assume o valor como 1 após o intervalo comparador especificado, e depois volta a zero à meia-noite.



- Para a onda quadrada, pulso e comparador, se você definir a tag startValue, a tag alterna entre 0 e startValue (em vez de 0 e 1).

Tags de Referência

Tags com tipo Referência permitem o endereçamento dinâmico das variáveis.

O conceito é mais próximo ao conceito de Referências em programação dot NET, do que o conceito dos antigos ponteiros C + +. A razão é que um Tag de Referência, como as referências NET, são Tipificados. O que significa que, quando da criação de um Tag de Referência, você precisa definir a que Tipo de tag ou de Templates de Tag a Referente estará apontando. Como as referências dot NET ao contrário dos ponteiros de C + + um Tag de Referência não pode ser definido para apontar para áreas inválidas de memória que poderiam causar erros na aplicação.

O tipo do tag que o Tag de Referência aponta é definido na coluna **parâmetros** da tabela de **Objetos.**

O fato de que ele é uma referência tipificada traz vantagens tanto em ambiente de Engenharia como no de "Run Time". Em ambiente de Engenharia permite a facilidade de Intellisense para navegar diretamente aos membros de um Template, quando a referência aponta para um Template; No tempo de execução permite a validação de strings de caracteres.

Todos os Tags de Referência tem um atributo adicional em tempo de execução que é o **link (elo),** que contém, em tempo de execução o Tag para o qual a Referência está apontando.

Essencialmente a propriedade da ligação é uma propriedade do tipo string que deve receber o nome do tag de destino antes que se use o tag de Referência. Você pode atribuir uma string diretamente ou usando uma expressão de strings, mas a melhor maneira de definir a propriedade "Link" é de usar o método getName (), que irá criar a string com o nome do Tag atual. Dessa forma, você pode renomear um tag sem ter que procurar strings de caracteres fixos em códigos de scripts. Esta forma também permite que o utilitário de Referência Cruzada, mostre os nomes de tags 'linked" (ligados).

Um exemplo de projeto de código usando Tags de Referencia é mostrado a seguir.

Exemplos

```
@Tag.Reference1.Link = @ Tag.TagName.GetName () (VB)
```

```
@Tag.Reference1.Link = @ Tag.TagName.GetName (); (C #)
@Tag.Reference1.Link = "Tag.TagName" (C #)
@Tag.Reference1.Link = "Tag.TagNa" + "me"; (C #)
```

NOTA - A razão para usar o métodos getName () em vez de usar diretamente os strings constantes, mesmo que seja uma sintaxe permitida, é que quando se utiliza o método getName () você mantém os benefícios de Referência-Cruzada e Refatoração, que não seria possível se você usasse atribuições diretas de cadeias de caracteres.

Formatos de Tags

Para conhecer melhor os formatos numéricos válidos consulte Standard Numeric Format Strings (http://msdn.microsoft.com/en-us/library/dwhawy9k%28v=VS.90%29.aspx). Por exemplo: N1 (numero com 1 casa decimal).

Para formatos válidos de datas ou horários, consulte Standard Date and Time Format Strings (http://msdn.microsoft.com/en-us/library/az4se3k1%28v=VS.90%29.aspx). Por exemplo: d (spara o formato data abreviada).

Para uma discussão mais aprofundada sobre strings de Formatos consulte Formatting Types (http://msdn.microsoft.com/en-us/library/fbxft59x%28v=VS.90%29.aspx).

Veja a tabela a seguir para exemplos de formatos.

Exemplos de formatos numéricos

Formato	Descrição
N0	Numero sem casas decimais
N3	Numero com 3 casas decimais
X	Hexadecimal (suportado apenas para números inteiros)
C	Moeda

Veja a tabela para exemplos de formatos para datas e horários.

Exemplos de formatos para datas e horários

Formato	Descrição
T (only)	Padrão de horário longo (equivalente à HH:mm:ss).
d (only)	Padrão de horário abreviado (equivalente à M/d/yyyy (month/day/year) para en-us).
dd	Mostrar o dia do mês como um número de 01 ATÉ 31.
ddd	Mostrar o nome abreviado do dia da semana.
dddd	Mostrar o nome completo do dia da semana.
MM	Mostrar o mês como um número de 01 ATÉ 12.
MMM	Mostrar o nome abreviado do mês

Exemplos de formatos para datas e horários

Formato	Descrição
yy	Mostrar o ano como um número de dois dígitos.
yyyy	Mostrar o ano como um número de quatro dígitos.
hh	Mostrar a hora como um número de 00 ATÉ 12.
HH	Mostrar a hora como um número de 00 ATÉ 23
mm	Mostrar o minuto como um número de 00 ATÉ 59.
ss	Mostrar os segundos como um número de 00 ATÉ 59.
fff	Mostrar os milissegundos como um número de 000 ATÉ 999.
tt	Mostrar os designadores A.M. / P.M.

Criando Tipos Customizados (Tag Templates)

Além dos tipos nativos (buil-in) simples pode se ter outros tipos definidos pelo usuário para implementar modelos de dados mais complexos, Estes tipos criados pelo usuário são denominados Templates, como protótipos a serem utilizados na aplicação.

Por exemplo pode-se criar agrupamentos de tags de tipos simples, de modo a criar-se tags que sejam correspondentes não apenas a um ponto monitorado mas a grupos de pontos que constituem os dados de um objeto real, como por exemplo um disjuntor, ou um "bay" de uma subestação, como um alimentador. Para esta definição se usa a aba Templates no espaço de trabalho dos Tags.

Para criar um Tag Template:

1. Vá para **Edit > Tags > Templates**. .
2. Clique **New** .
   - Aparece a janela Create New Tag Template.
3. No campo New type name, digite um nome para o novo tipo de tag. Em Description, digite uma descrição do tag. Clique em **OK**.
4. A aba Templates é mostrada com o nome do novo Template na parte superior da aba. O grid abaixo será limpo para a entrada dos tags que vão compor este Tipo.
   - Clique na linha de inserção para criar um novo atributo para esse Template de tag.
5. Digite ou selecione as informações, conforme necessário.

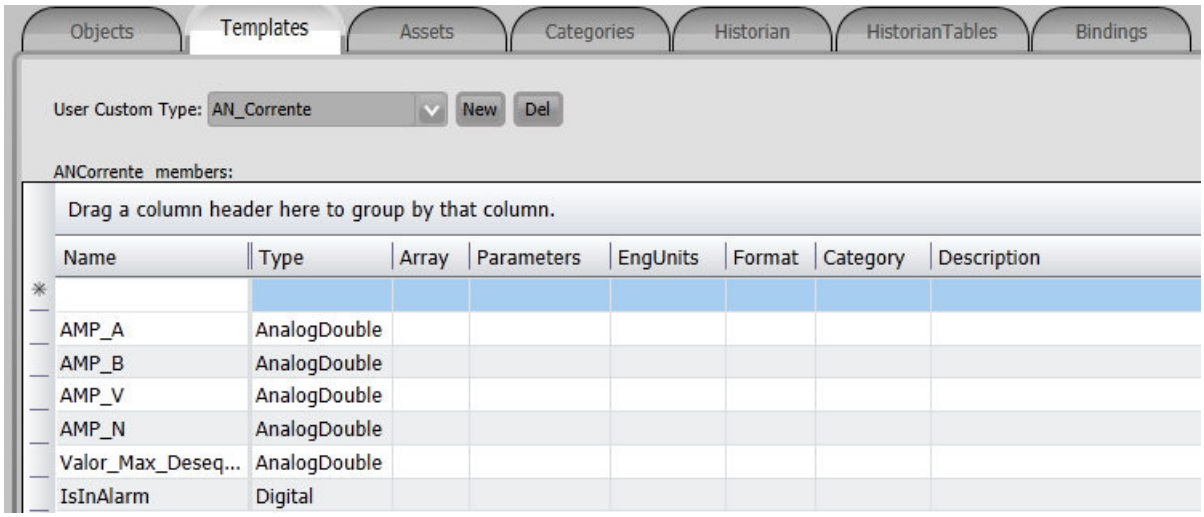
Coluna	Descrição
Name	Digite um nome para o atributo. O sistema fará om que você saiba se o nome não é válido. pelo contorna da célula em vermelho.
Type	Selecione o tipo de variável. Escolha um dos "Tipos de Tags pré-definidos" na pagina 82 ou um dos Templates de tag previamente definidos.

Coluna	Descrição
Parameters	Se estiver disponível, configure os parâmetros. Os parâmetros variam conforme o tipo de Tag.
Array	Se você precisa de uma matriz, defina último índice de matriz. Por exemplo, se você digitar 5, o sistema cria um Tag com seis elementos (0-5).
Description	Digite uma descrição para este atributo.
[Outras colunas]	As colunas na tabela Templates tem o mesmo uso das colunas da tabela Os membros têm o mesmo uso das colunas em EditTagsObjects.

- Para excluir um Template, selecione-o na lista drop-down na parte superior da aba, "User custom type" e clique em Del .
- Na aba **Objects** para os novos tags ou os já existentes, você pode a partir de então usar este novo Template escolhendo-o na coluna Type.

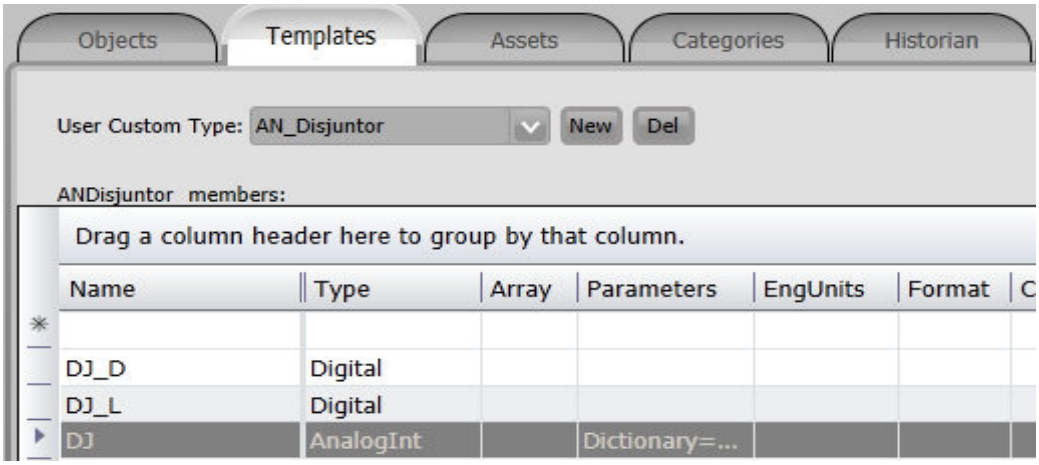
Utilizando Tipos Customizados (Templates)

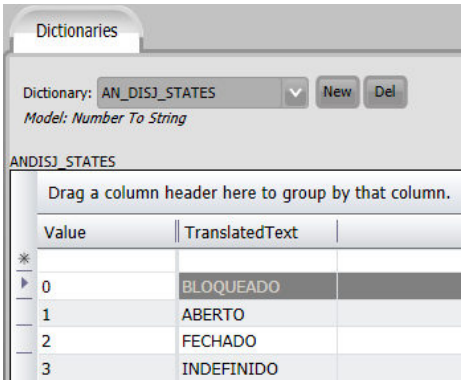
Veja no exemplo a seguir: é mostrado um Tipo Customizado ou Template denominado **AN_Correntes**, para agrupar em um único objeto as três fases da corrente mais o percentual máximo de desequilíbrio de fase, e um tag digital que será utilizado para indicar alarme se o desequilíbrio calculado for superior a este máximo.



Veja outro Tipo: No caso os tags necessários para a monitoração e comando de um disjuntor, com o nome **AN_Disjuntor**:

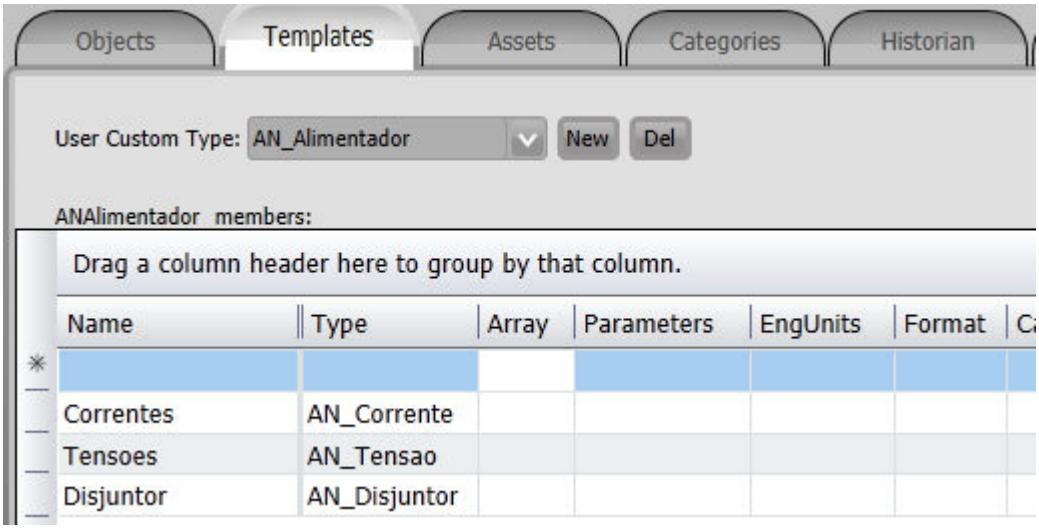
- Inclui um tag simples analogInt, que conterá a codificação dos estados possíveis do disjuntor (de dois contatos);
- Dois tags que terão como atributos os endereços para o envio de comandos de abrir e fechar o disjuntor.





No parâmetro do tag de sinalização DJ, há a referência a um **Dicionário (Enumerator dictionary)** que vai conter os textos a serem mostrados para cada estado de DJ (Para criar dicionários veja em "Dicionários e Localização" na pagina 71).

Veja agora mais um tipo, no caso um que define um objeto Alimentador, com sua medição e um disjuntor com nome **AN_Alimentador.** Ele foi criado utilizando os tipos **AN_Correntes e AN_Disjuntor** já mostrados acima. Portanto foi criado aqui um novo Template composto por um agrupamento de pontos que pode servir para monitorar inteiramente um objeto real Alimentador., que usa como atributos outros Templates



Agora para se criar um objeto Alimentador, deve-se ir para Objetos, e inserir um novo tag, por exemplo: SE_DEMO_ALIM_01 e escolher na lista de tipos o **AN_Alimentador.** Com este procedimento todos os tags para o disjuntor e a medição serão criados automaticamente. Os nomes dos tags terão os nomes dos objetos de tipo constituintes como:

SE_DEMO_ALIM_01.Correntes.AMP_A

SE_DEMO_ALIM_01.Correntes.AMP_B

SE_DEMO_ALIM_01.Disjuntor.DJ

SE_DEMO_ALIM_01.Disjuntor.DJ_L

Criando Níveis (Assets)

Se você tiver a versão Enterprise do Action.NET os Níveis ou (Assets) permitem configurar metadados adicionais para o seu projeto. Por exemplo, você pode organizar os objetos em seu projeto, tais como Tags, Devices e Alarmes, em uma hierarquia. Isto permite-lhe agrupar Tags que estão relacionados entre si.

Nas concessionárias de energia, usa-se, geralmente, para se identificar um "bay" a subdivisão do sistema elétrico em *Regionais*, *Subestações e Setores de tensão.* Os objetos níveis existem justamente para esta definição hierárquica.A hierarquia pode refletir coisas como, UHEs, Subestações, Setores de tensão.

Define-se na aba **Níveis (Assets)** do espaço de trabalho dos Tags

Para criar Níveis (Assets):

1. Vá para **Edit > Tags > Assets**.
2. Botão direito do mouse no nome do projeto e selecione **New Level**
3. Digite um nome para o nível.
4. Clique com o Botão direito do mouse o novo nível e selecione **Insert Asset** .
   - A janela Selecionar objeto é exibida, com todos os objetos, por tipo, no lado esquerdo.
5. Selecione o tipo de objeto a partir do lado esquerdo e do objeto que você deseja a partir do lado direito.
6. Clique em OK.
   - O objeto torna-se um filho do nível selecionado.
7. Continue adicionando níveis filho ou irmão e inserção de ativos, conforme a necessidade.
   - Se necessário, clique com o botão direito um nível para renomear ou apagar, ou clique com o botão direito em um Nível para excluí-lo.
8. Na aba **Objetos**, para Tags novos ou já existentes, selecione o novo nível na coluna Level.

Exemplo do uso de Níveis

No exemplo da figura a seguir há, como primeiro nível as UHEs AGV, ILS, COS, etc. que pertencem ao projeto DEMO, e como segundo nível da UHE ILS o setor de máquinas ILS_UHE. Ao se cadastrar os tags, no campo nível (Level) deve-se preencher com um destes níveis ao qual o tag pertence.

Uma vez criado, o Nível ficará disponível na lista de existentes na coluna Level (Nível) da aba Objeto, para a criação de Tags.

Criando Categorias

Se você tiver a versão Enterprise do Action.NET, você pode criar categorias de dados, definidas pelo usuário, que você pode usar como metadados para tags. As categorias são úteis para filtragem, tanto no ambiente de Engenharia como em tempo de execução.

Para criar categorias:

1. Vá para **Run > Dictionaries > Categories** .
2. Digite ou edite o nome e a descrição para a categoria.
3. Continue adicionando tantas categorias como você precisa.
4. Na abas **Tag -> Objetos**, para Tags novos ou já existentes, selecione a nova categoria na coluna Categoria. Outros elementos do projeto também podem usar categorias para organização do projeto.

Para um exemplo do uso de Categorias veja "Dicionário de Categorias" na pagina 74.

Objetos Tags em tempo de execução

Tag Namespace

Todas as variáveis do projeto em tempo real, ou tags do projeto como eles geralmente são referidas no contexto de automação de processos, criados na tabela de Objetos Tags , estão disponíveis para os módulos de tempo de execução como objetos dot NET no **Namespace Tag**.

Todos os tipos buil-in para Tags, compartilham um conjunto comum de propriedades e métodos definidos na classe base TagObj . Os Tags criados a partir de Templates definidos pelo usuário, são implementados pela classe base **`UserType`**. Veja em "Criando Tipos Customizados (Tag Templates)" na pagina 85.

Classe tipo	Descrição
`TagObj`	`Classe Base para todos os objetos Tags.`
`Digital`	Propriedades em tempo de execução para as tags do Tipo Digital.
`Analog`	Propriedades em tempo de execução para todos os tag do tipo analógico.
`AnalogInt`	Propriedades em tempo de execução para as tags do tipo inteiro.
`AnalogLong`	Propriedades em tempo de execução para as tags do tipo long.
`AnalogDecimal`	Propriedades em tempo de execução para as tags do tipo decimal.
`AnalogDouble`	Propriedades em tempo de execução para as tags do tipo double.
`Text`	Propriedades em tempo de execução para as tags do tipo texto.
`TDateTime`	Propriedades em tempo de execução para as tags do tipo DateTime..
`Timer`	Propriedades em tempo de execução para as tags de tipo de temporizador.
`TTimeSpan`	Propriedades em tempo de execução para as tags do tipo TimeSpan.
`Reference`	Propriedades em tempo de execução para as tags de tipo Referência.
`TDataTable`	Propriedades em tempo de execução para as tags do tipo DataTable.
`UserType`	Propriedades em tempo de execução para as tags Templates

Veja em http://www.spinengenharia.com.br/help/an-2014/runtime/index.html para uma referência completa sobre programação usando objetos em de tempo de execução.

Historiador e Registro de dados

O módulo historiador faz o registro de dados automatizado para bancos de dados do tipo SQL e outros. Também se pode usar o módulo de Datasets para armazenar dados em SQL, o módulo Historiador porém tem uma configuração simplificada, com as tabelas de banco de dados sendo criadas automaticamente e uma conexão já pré definida com os gráficos de tendência.

O sistema também pode usar as informações disponíveis na base de dados do historiador, em Scripts dot NET feitos pelo usuário, para obter valores históricos em Tags e mostrá-los em telas gráficas ou para exportar estes valores.

As seções a seguir descrevem como configurar o registro de dados:

- "Adicionando Tags para o Historiador" na pagina 91.
- "Configurando Tabelas do Historiador" na pagina 92.
- "Provedor do banco de dados Historiador" na pagina 93
- "Usando o Sistema PI da OSIsoft(tm)" na pagina 94
- "Schema de Tabelas do Historiador" na pagina 94.
- "Visualizando Gráficos de Tendência" na pagina 95.
- "Personalizando a Obtenção de Amostragens" na pagina 95.
- "Objetos runtime do Historiador" na pagina 95

Pode-se selecionar qualquer banco de dados SQL, como o Microsoft SQL Server, Oracle, MySQL ou qualquer OLEDB ou banco de dados compatível com ODBC para o armazenamento de dados históricos. Por default o Action.NET utiliza um "engine" de banco de dados SQL embutido (SpinDB), que tem capacidade máxima de 10 GB. Para obter informações sobre como selecionar o banco de dados do historiador, consulte o item "Provedor do banco de dados Historiador" na pagina 93.

É possível também a utilização do sistema PI OSIsoft (tm), para o armazenamento de dados. Neste caso não há necessidade de fazer qualquer configuração de Historiador para acessar os dados armazenados. Para detalhes sobre o uso do Sistema PI, consulte "Usando o Sistema PI da OSIsoft(tm)" na pagina 94.

Adicionando Tags para o Historiador

Depois de configurar suas tabelas para o historiador de tags, você pode configurar as tags cujos valores deverão ser registrados pelo historiador.

Para configurar as tags para o historiador:

1. Vá para **Edit > Tags > Historian**.
   - Você pode copiar e colar as Tags a partir da aba **Objects**.
2. Digite ou selecione as informações, conforme necessário.

Coluna	Descrição
Name	Digite um nome de tag ou clique para selecionar um tag.
DeadBand	Ao usar a opção Save On Change na tabela historiador, a DeadBand é o quanto o valor deve mudar (em EngUnits) para disparar no sistema a gravação do valor no historiador.
Deviation	Ao usar a opção Save On Change na tabela historiador, o desvio é o quanto o valor do Tag deve mudar (em EngUnits) para o sistema armazenar um novo valor no historiador. Esse valor substitui o intervalo de tempo de registro Time DeadBand .
RateOfChange	Ao usar a opção Save On Change na tabela historiador, o RateOfChange é quanto o valor do Tag deve mudar (em EngUnits) por segundo, para o sistema armazenar um novo valor no historiador. Esse valor substitui o intervalo de tempo de registro Time DeadBand .
HistorianTable	Selecione a tabela que tem as configurações que você deseja usar para periodicidade de registro e por quanto tempo manter valor deste tag

3. Continue adicionando quantos Tags quantos você precisa.

Configurando Tabelas do Historiador

Por default, o banco de dados do Historiador de tags já possui uma tabela configurada. A configuração desta tabela controla quando os valores dos tags devem ser escritos para o historiador e por quanto tempo estes valores devem ficar retidos no historiador. Se você quiser que alguns tags tenham tratamento diferenciado, você deve configurar tabelas adicionais com as definições que necessita e atribuir os tags para alguma destas tabelas. Você também pode editar as configurações da tabela padrão, se achar conveniente.

A configuração da tabela é independente do tipo de banco de dados selecionado para armazenar as informações, conforme descrito em "Provedor do banco de dados Historiador" na pagina 93.

Em geral, não armazene mais dados do que você precisa. Armazenar uma grande quantidade de dados reduz o desempenho na recuperação de dados. Você deve usar gatilhos e bandas mortas tão amplos quanto possível, para garantir que terá a informação que você realmente precisa, sem sobrecarregar o sistema.

Para configurar uma tabela historiador:

1. Vá para **Edit > Tags > Historiador**.
2. Siga um dos seguintes procedimentos:
   - Para editar uma tabela existente, selecione-a na lista drop-down Tabelas Historiador e clique **Config**.
   - Para criar uma nova tabela, clique **New**.

Opção	Descrição
Database	Mostra o nome do banco de dados do Historiador atual.(somente leitura)
Table Name	Digite um nome para a tabela no banco de dados
Auto Create	Selecione para que o sistema crie automaticamente a tabela no banco de dados.
Save on Change	Selecione para armazenar dados na tabela (adicionar uma linha) cada vez que um tag associado à tabela sofrer modificação.
Trigger	Use para registrar dados na tabela cada vez que um tag ou propriedade do tag sofre mudanças. Quando usado com a opção Save on Change, o sistema armazena dados na tabela tanto quando o valor do tag muda como quando ou o valor de Trigger se modifica
Time Deadband (Log TimeSpan)	Entre com o mínimo intervalo de tempo entre gravações. Isto é quanto o sistema deve esperar depois de gerar um registro para um tag antes de gerar um novo registro para o mesmo tag. Use com a opção Save on Change para evitar a geração de registros em demasia na base de dados.
Life Time	Numero de dias em que devem ser retidos os dados históricos. Depois deste tempo, as linhas mais antigas na tabela serão excluídas da base de dados. Para que os dados nunca sejam excluídos deixe este campo vazio ou digite um zero.
Compress	Numero de dias ante de fazer uma compressão dos dados. Os dados comprimidos ficam ainda acessíveis, mas o acesso é mais lento. Para que nunca seja feita compressão deixe este campo vazio ou digite um Zero.

Provedor do banco de dados Historiador

O banco de dados usado para armazenar o Historiador é definido em **Edit > Datasets > DB**s pelo objeto de conexão à banco de dados com o nome de **TagHistorian**.

Por default, quando um novo projeto é criado, este registro de nome TagHistorian é definido para usar o banco de dados **SQL SpinDB** que está embutido dentro do Action.NET

Cuidado: O banco de dados **SQL SpinDB** deve ser usado para bancos de dados de até 10 GB. Se a quantidade de tags e periodicidade de salvamentos exigir armazenamento de mais de 10 GB, você deve definir outro sistema de banco de dados SQL para o Historiador. Para definir um outro banco de dados para o Historiador, você só precisa criar uma nova conexão de banco de dados, como explicado em "Configurando Conexões de Bancos de dados" na página 127, e nomeá-la TagHistorian.

Dica: O sistema não permite nomes duplicados, deste modo para criar uma conexão DB com o nome TagHistorian você deve renomear ou excluir a linha existente usando esse nome.

Usando o Sistema PI da OSIsoft(tm)

O Action.NET pode usar o Sistema PI OSIsoft como provedor para o Historiador. Neste cenário, não é necessário fazer qualquer configuração adicional. Se o Tag é mapeado para um ponto OSIsoft PI, o sistema irá automaticamente chamar o PI Server para obter dados ao plotar gráficos de tendências ou quaisquer outros métodos de scripts ou de Tela que requeiram informações do Historiador.

O sistema pode funcionar com o Sistema PI e o Historiador embutido, ao mesmo tempo. Quando a informação histórica de um ponto é solicitada, a partir de gráficos de tendências ou scripts, o sistema irá procurar estas informações no módulo Historiador embutido e, se não encontrar, vai tentar encontrar esses dados no servidor PI.

Para obter informações sobre a ligação a sistemas PI, consulte o item "Assistentes de Importação" na pagina 75.

Schema de Tabelas do Historiador

As tabelas do Historiador contém as seguintes colunas:

- *UTCTimeStamp_Ticks*- Data e hora em UTC (GMT) para este registro (em formato 64-bits NET ticks). O valor dessa propriedade é o número de intervalos de 100 nanossegundos (1/10th de um milésimo de segundo) que se passaram desde 00:00, 1 de Janeiro de 0001. Este é um novo padrão de data / hora usado pelo Microsoft dot NET framework..
- *LogType*- Coluna auxiliar para mostrar quando a linha foi inserida: 0 = na partida, 1 = como registro normal, 2 = no encerramento do sistema.
- *TagName* - Coluna criada automaticamente usando o nome da tag, como título da coluna. Ela armazena o valor do dado usando precisão dupla. Haverá uma coluna destas para cada Tag definido para esta tabela
- *TagNameQ* - Coluna criada automaticamente para a qualidade dos dados, usando a especificação de qualidade OPC.Haverá uma coluna destas para cada Tag definido para esta tabela

Normalmente, você pode associar até 200 tags com cada tabela do historiador, mas esse número é dependente de quantas colunas o banco de dados sendo utilizado permitir. Os Tags devem ser definidos na mesma tabela quando eles têm critérios de gravação, intervalos de tempo, taxas de variação e dinâmica no processo, semelhantes, já que para salvar uma tag na tabela, você precisa salvar a linha inteira.

Visualizando Gráficos de Tendência

Para visualizar gráficos de tendências com informações históricas, você pode usar o objeto de tendência disponível no espaço de Engenharia utilizado para definição de telas ou relatórios. Consulte a seção "Configurando a Janela de Tendência" na página 179 para mais informações.

Personalizando a Obtenção de Amostragens

Normalmente o objeto de tendência gráfica chama o servidor Historiador para obter os dados para plotar os gráficos. Em algumas situações, você pode querer substituir essa configuração e definir um código script em dot NET para obtenção dos valores. Isto é usado, por exemplo, para plotar dados calculados para uma receita, ou dados futuros, ou ainda dados de outras tabelas SQL ou qualquer cenário em que haja necessidade de uma personalização na obtenção de dados.

Para esta personalização um método `GetSamples` deve ser definido, qualquer Classe de scripts, para fazer a obtenção de amostragens. Na aba **Edit>Tags>Historian** este método deve ser definido na coluna `GetSamplesMethod`.

O protótipo para este método é:

```
DataTable GetSamples(string[] tagNames, object startRange, object
endRange)
```

Os parâmetros:

- Os parâmetros startRange and endRange são do tipo:
  - dotNET DateTimeOffset, quando utilizados para obter tabelas usadas em gráficos de tempo.
  - Double, quando criando dados para gráficos X-Y.
- O objeto retornado pelo método DataTable deverá ter as seguintes colunas:
  - DateTime: A data e hora da amostragem
  - TagName: O nome do tag utilizado como FieldName para a coluna com o valor do tag em Double.
  - TagName_Q: Coluna opcional com a qualidade do dado na codificação OPC.

Objetos runtime do Historiador

O namespace **`Historian`** tem as propriedades e o estado atual do servidor Historiador.

O objeto **`Historian.Table`** tem a lista de tabelas definidas no historiador e as propriedades. de cada tabela.

A propriedade de tag, `Tag.tagname.Historian`, fica habilitada para os tags, se há definição de registro histórian no projeto.

Segurança e Usuários

Por default, ao iniciar a execução do Action.NET o sistema registra a seção como sendo de um usuário pré definido “Guest”. Este usuário (convidado) não precisa digitar um nome de usuário ou senha.

Se você quiser garantir o acesso a seus projetos, você pode adicionar usuários e configurar senhas e permissões diferenciadas para cada usuário.

As seguintes seções descrevem como configurar a segurança do sistema e projeto.:

- “Adicionando e Alterando Usuários” na pagina 97.
- “Configurando Permissões” na pagina 98.
- “Configurando Políticas” na pagina 99.
- “Removendo Usuários” na pagina 100.
- “Gerenciando usuários em Tempo de execução” na pagina 101.
- “Personalizando procedimentos de Login” na pagina 101.
- “Objetos runtime de Segurança” na pagina 102

Adicionando e Alterando Usuários

Para aumentar a segurança dos dados de uma instalação com o Action.NET, você pode adicionar e editar usuários. O sistema ao ser instalado vem originalmente com os seguintes nomes de usuário configurados por default:

- Administrator - usuário que controla o sistema de segurança. Nenhuma senha é configurado por padrão. Você deve definir uma senha para este usuário.
- Guest- Usado por default para acessar o Action.NET e quando terminar a sessão com outro usuário. Nenhuma senha é configurada por default.
- User - Usado como um usuário genérico. Nenhuma senha é configurada por default.

Os níveis dos Usuários com identificação de Administrator e Guest tem atributos pré definidos no sistema. O Administrator é o unico usuário que pode excluir ou alterar permissões de um usuário e definir senhas para interfaces de bancos de dados na tabela DBs. O usuário Guest é utilizado para o acesso anonimo e não tem senha associada a ele. Não tem também permissão para aletrar configurações no espaço de Engenharia.

Não crie outros usuários com estes nomes, nem altere as configurações destes registros na tabela de usuários, porque o sistema presume estes nomes com estes atributos.

Nota: Não se pode excluir o usuário Guest, e nem adicionar uma senha para ele. O usuário Guest deve sempre estar disponível como um usuário default para o qual o sistema se auto-loga, quando outro usuário faz um “log off”. Você pode querer alterar as permissões do usuário Guest, para que não tenha acesso a todas as funções. Veja “Configurando Permissões” na pagina 98.

Para adicionar e editar usuários:

1. Vá para **Edit > Security > Users** .
2. Digite ou selecione as informações, conforme necessário.

Coluna	Descrição
Name	Digite um nome de usuário. O sistema fará com que você saiba se o nome não é válido, mostrando a célula com o contorno vermelho
Permissions	Selecione o grupo de permissão de uso para este usuário. Veja "Configurando Permissões" na pagina 98.
Password	Digite uma senha para o usuário. O sistema permite que você saiba se a senha não é válida. Você pode configurar requisitos de senha. Veja "Configurando Políticas" na pagina 99.
PasswordHint	Entre as informações que podem ajudar você a lembrar a senha do usuário.
Policy	Selecione as configurações de política de usos para este usuário. Veja "Configurando Políticas" na pagina 99.
Blocked	Selecione bloquear o acesso do usuário. Pode-se querer usar isso para usuários que não estão mais na empresa.
Deleted	Selecione para bloquear o acesso e marcar o usuário como excluído, sem excluir o usuário. Você pode querer usar isso para usuários que não estão mais na empresa.
Profile	Digite o endereço de e-mail, número de telefone e nome completo do usuário

3. Continue adicionando tantos usuários quanto você precisar.

Configurando Permissões

Você pode configurar as funções que os Usuários terão permissão de acesso ao editar o projeto e quando estiverem utilizando o sistema em tempo de execução. Você define permissões usando o que é denominado por Grupos de Usuários.

O Action.NET vem com alguns Grupos predefinidos que você pode usar como estão. Você pode também criar outros grupos conforme sua necessidade.Cada Grupo utiliza uma linha da tabela.

Para configurar permissões:

1. Vá para **Edit > Security > Permissions**.
2. Digite ou selecione as informações, conforme necessário.

Coluna	Descrição
Name	Digite um nome para o grupo. O sistema fará com que você saiba se o nome não é válido, mostrando a célula com o contorno vermelho.
Edit	Clique na célula. Aparecerá uma lista de itens de Menus existente no espaço de Engenharia. Selecione as funções que os usuários neste grupo podem acessar durante a edição de um projeto.
Run	Clique na célula. Aparecerá uma lista de itens de Menus existente no espaço de Engenharia e funções de tempo real. Selecione as funções que os usuários neste grupo podem acessar durante a utilização do sistema em tempo de execução
Description	Digite um texto descritivo sobre o grupo

3. Continue adicionando quantos grupos você precisa.
   - Os grupos de permissões que você criar estarão disponíveis para uso na aba **Users**.

Desabilitar opção Alternar Aplicativos do Windows

Uma das permissões de tempo de execução é desativar a facilidade de alternar de um aplicativo para outro no ambiente do MS-Windows. A função é `EditSecurityPermissions.Run`. Se desativada, as teclas e ações, a seguir listadas, ficarão inibidas, durante a execução em tempo real do Action.NET:

- O acesso ao botão Iniciar e a Barra de Ferramentas de tarefas.
- botão com o logotipo do Windows
- Ctrl + Alt + Del
- Ctrl + ESC
- Alt + Tab
- alt + F4
- Logotipo do Windows + L

NOTA - Para permitir desativar os atalhos de teclado que alternam entre aplicações, deve ser executado o arquivo **DisableTaskSwitchProtection.bat** a partir do diretório de instalação. Esta ação serve para instalar o driver de teclado necessário para implementar esse recurso.

Configurando Políticas

Pode-se configurar as políticas de permissão que se desejar aplicar aos usuários.

O Action.NET, por default, já possui algumas políticas predefinidas que podem ser usadas. Se preferir, você também pode criar outras, conforme sua necessidade.

Para configurar políticas:

1. Vá para **Edit > Security > Policies**.
2. Digite ou selecione as informações, conforme necessário.

Coluna	Descrição
Name	Digite um nome para a política. O sistema fará com que você saiba se o nome não é válido, mostrando a célula com o contorno vermelho.
Identification	Selecione as regras de senha, tanto para a edição de projetos como para acessar projetos em tempo de execução.
Esign	Apenas para tempo de execução. Selecione a opção se desejar impor um tempo limite para a sessão runtime. Digite o período de tempo limite em minutos.
Session	Para tempo de execução somente. Selecione se vai causar um logoff automático, em seguida, digite os valores apropriados para InactivityMinutes e DurationHours. Esta definição somente faz o encerramento da sessão do usuário. O aplicativo continua a ser executado.
Description	Digite uma descrição para a política.

3. Continue adicionando tantas políticas quantas você precisa.
   - As políticas criadas estão disponíveis para uso na aba **Users**
   - Se necessário, clique com o botão direito em uma linha da tabela para cortar, copiar, colar ou excluir a linha.

Removendo Usuários

Você tem três maneiras de desativar usuários em Action.NET:

- Bloqueio - Use para bloquear o acesso do usuário. Pode-se usar esta opção para impedir o acesso de usuários que deixaram a sua empresa.
- Sinalização de Excluídos - Use para bloquear o acesso do usuário e sinalizar o usuário como excluído, sem excluir o usuário. Você pode querer usar isso para usuários que não estão mais na sua empresa.
- Excluindo - Remove o usuário completamente do sistema.

Você deve usar o método que é melhor para a situação da sua empresa.

Para remover usuários:

1. Vá para **Edit > Security > Users**.
2. Clique no usuário que você deseja remover.
3. Siga uma das seguintes opções:
   - Para bloquear um usuário, selecione a coluna Bloqueado.
   - Para sinalizar um usuário como excluído, selecione a coluna Excluídos.

- Para apagar completamente um usuário, clique com o botão direito do mouse na linha e selecione **Delete selected row**..

Gerenciando usuários em Tempo de execução

Ao executar o aplicativo, o procedimento de login vai aceitar todos os usuários definidos na configuração do projeto, mas usuários adicionais, definidos somente pelo sistema podem também ser incluídos.

Os usuários de tempo de execução são definidos no banco de dados **RuntimeUsers** definidos no menu **Datasets.** Por default, esse banco de dados é mapeado para um banco de dados SpinDB. Você pode selecionar qualquer outro banco de dados através da criação de uma outra conexão DB e atribuir a ele o nome RuntimeUsers (você terá de apagar ou renomear a linha RuntimeUsers existente, antes de criar o nova, pois a tabela não aceita nomes duplicados).

Você pode preencher esse banco de dados diretamente antes de iniciar a execução do sistema, ou depois que o projeto já está sendo executado, você pode criar ou modificar os usuários, através de scripts em dot NET, com os métodos disponíveis no Namespace de Segurança.

Veja em http://www.spinengenharia.com.br/help/an-2014/runtime/index.html para a referência de programação completa sobre objetos runtime (de tempo de execução).

Quando esses métodos são executados, os usuários novos ou modificados são automaticamente salvos no banco de dados RuntimeUsers.

Personalizando procedimentos de Login

A página de login é editável, você só precisa selecionar a tela com o nome de LogOn para editar o seu layout. Como você pode ver no código por trás dessa Tela, ele chama o método `Security.Logon ()` para fazer a validação do usuário. Se você quiser executar qualquer outra validação do usuário, você só precisa modificar essa lógica chamando o seu próprio sistema de validação e, em seguida, chamar o método de LogOn, de acordo com os resultados de sua validação.

Outra maneira de personalizar o logon é colocar sua própria lógica personalizada na tarefa de script `ClientStartup`. O script é executado em qualquer computador que está se conectando ao servidor de aplicação, você pode executar verificações baseada em IP do computador, nome do computador, se o usuário está logado no Windows Active Directory , ou qualquer outro critério para determinar se o usuário tem permissão para iniciar o aplicativo e quais deveriam ser as suas credenciais. Depois desta verificação pode chamar o procedimento `client.Shutdown`, para encerrar o aplicativo caso o usuário não for autorizado, ou `Security.Logon ()` se o usuário corresponder ao perfil de segurança selecionado.

Por default, quando se inicia o aplicativo em um computador cliente, em vez de solicitar um login, o sistema faz a inicialização da sessão com o usuário GUEST. O usuário Guest é equivalente a um login anônimo, se você não quiser este método em sua instalação, basta substituir a página inicial por uma página solicitando as informações de logon.

Objetos runtime de Segurança

O namespace **`Security`** tem todas as informações de tempo de execução em relação ao sistema de segurança.

O objeto **`Client`** tem informações sobre o usuário atual conectado nessa estação cliente:

`A` propriedade **`Client.Username`** contém o nome do usuário logado.

`Client.CurrentUser` é uma referência a uma estrutura de dados com todas as informações do usuário conectado no momento.

Veja em http://www.spinengenharia.com.br/help/an-2014/runtime/index.html para a referência de programação completa sobre objetos runtime (de tempo de execução).

Dispositivos (Devices) e Interfaces

Dispositivos ou IEDs (Inteligent Eletronic Devices) no Action.NET são quaisquer fontes de dados em tempo real. Tipicamente, um dispositivo é um PLC ou um Relé digital, um outro projeto Action.NET em execução, um servidor de OPC, um Sistema PI, ou qualquer equipamento que tenha um protocolo de comunicação para a troca de informações com o projeto.

As seções a seguir explicam como configurar o módulo de dispositivo, menu **Devices** do menu principal **Edit**:

- "Protocolos de Comunicação" na pagina 103 apresenta os protocolos e interfaces que são suportados.
- "Configurando Canais" na pagina 105 explica a configuração da aba Channel, ou canal é o nome usado pelo Action.NET para definir uma conexão a uma fonte de dados, por uma rede lógica executando um protocolo selecionado
- "Configurando Nós" na pagina 107. Nós ou nodos são as estações lógicas conectadas a cada canal, isso significa um mapeamento com os dispositivos que estão conectados a cada rede lógica.
- "Configurando endereços de pontos" na pagina 109 define o comportamento e o endereçamento para leitura ou escrita de pontos de dados em cada nó do dispositivo que será acessado;
- "Configurando Tipos de Acessos" na pagina 111 permite personalizar a forma como os pontos de dados serão acessados, como ler ou escrever, os ciclos de execução e gatilhos para execuções de tarefas.

Protocolos de Comunicação

Conectividade é uma característica fundamental na plataforma Action.NET, portanto, o sistema possui embutido suporte para vários protocolos padrão da indústria, tais como OPC e Modbus, além destes o Action.NET tem também muitas interfaces de comunicação nativas para uma grande variedade de protocolos proprietários de fabricantes de hardware e PLCs.

As razões para incluir protocolos nativos, além de OPC, são muitas, tais como: redução de custos, já que a maioria dos protocolos não são cobrados; configuração mais fácil, uma vez que está integrado no sistema; maior acesso a funções dos protocolos; por razões de desempenho e diagnósticos, características que não são possíveis quando se usa componentes externos como o OPC. O suporte técnico também é simplificado, pois o fornecedor dos protocolos é único e é o mesmo que desenvolve e fornece o SCADA.

Os seguintes protocolos de comunicação estão incluídos na distribuição padrão do produto: Listado por ordem alfabética

- Direção Automation, KOYO, usando o protocolo ECOM
- Mitsubishi MELSEC Q
- Modbus - TCP / IP e RS-232, RTU e ASC, protocolo mestre Modbus
- Modbus Slave - TCP / IP e, protocolo Modbus escravo RS-232, RTU e ASC

- OPC DA Cliente - OPC acesso do cliente aos servidores locais ou remotos
- Rockwell / AB ControlLogix
- Rockwell / AB MicroLogix
- Rockwell / AB DF1 Ethernet
- TRemoteClient - Action.NET a comunicação Action.NET
- TwinCAT - Interface Beckhoff ADS

Protocolos Suportados

Existem muitos protocolos que, embora não incluídos nos instaladores padrão do produtos, eles estão disponíveis mediante pedido e sem custos. Alguns desses protocolos já estão em estado de utilização em produção, outros estão em fase de desenvolvimento e testes. Entre em contato com a SPIN se você tem necessidade de algum dos protocolos nesta lista. Mesmo se o equipamento com o qual sua aplicação necessita comunicar-se não estiver listado, mas não há informações públicas sobre o protocolo, contate-nos já que, em muitos casos, desde que se possa ter acesso à documentação do protocolo e unidades físicas para testes, não há cobrança para criar interfaces nativas.

Em ordem alfabética:

- Altus AL-1000, AL2000 and Nexto PLCs
- Fatek Facon PLC
- GE Fanuc SNP and SNP-X
- Mitsubishi Series A and FX
- Omron CS/CJ/CP-series CPU Unit ou NSJ Controller
- Reliance CP3000
- Smar CD600
- SMTP - Simple Network Management Protocol

Interfaces adicionais

Há algumas interfaces que estão disponíveis e não foram incluídas na distribuição do produto, pois têm seus próprios pacotes de instalação ou são vendidos como módulos opcionais. Nestes estão incluídos os seguintes protocolos, em ordem alfabética:

- DNP 3.0 Protocol (mestre e escravo)
- IEC 61850 (somente cliente)
- IEC-870-5-101 (mestre)
- IEC-870-5-104 (mestre e escravo)
- OSIsoft PI System e PI AF, usando o conector AFSDK

Toolkit para protocolos

A equipe de desenvolvimento da SPIN Engenharia tem uma vasta experiência na criação de módulos protocolos de comunicação, que inclui mais de duas centena de interfaces criadas ao longo desse últimos vinte anos.

Resultado dessa experiência acumulada, o Action.NET tem uma interface padrão aberta, um Toolkit de Protocolos, para permitir adicionar novos protocolos de uma forma fácil e eficiente. Uma infra-estrutura padronizada embutida no módulos de execução de Devices cuida de tudo relacionado ao envio e recebimento físico das mensagens, o acesso as tags em tempo real, a configuração dos endereços e os diagnósticos e ferramentas de desempenho.

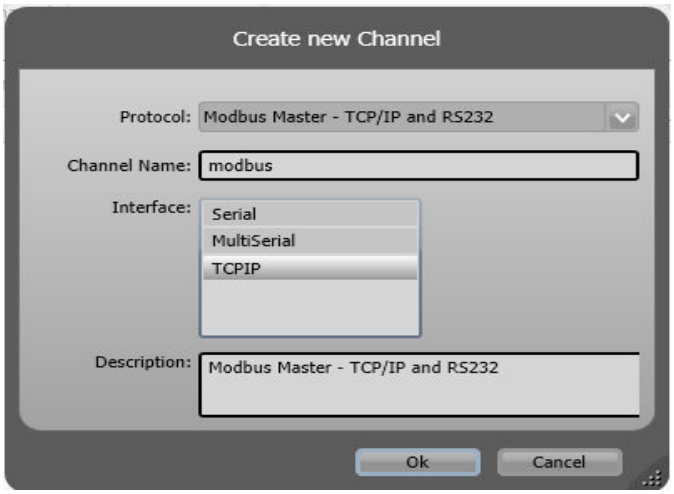
A ferramenta Toolkit de protocolos está disponível para parceiros de integração de sistemas e usuários finais licenciados. Entre em contato conosco se você tiver a necessidade de criar suas próprias interfaces personalizadas.

Configurando Canais

Canais em Action.NET são os objetos que implementam meios de comunicação utilizados para a comunicação do ambiente do SCADA com os PLCs e outros IEDs. Muitos protocolos internos estão disponíveis. Você deve configurar pelo menos um canal para cada protocolo que você precisa usar.

Para configurar canais:

1. Vá para **Edit > Devices > Channels**
2. Clique em **Create New**.
   - É mostrada a janela de criação de novo canal (figura a seguir)



   - Digite ou selecione as informações, conforme necessário.

Coluna	Descrição
Channel Name	Digite um nome para o canal. O sistema indicará se o nome não é válido, mostrando a célula com o contorno vermelho.
Protocol	Selecione o protocolo que vai utilizar neste canal. Se você não vê um protocolo que você precisa, clique em Mais para fornecer informações para um novo protocolo.
Interface	Selecione o tipo de interface para este canal. • Serial-Use para configurar os parâmetros seriais para redes RS232/485. • MultiSerial-Use para configurações com várias portos RS-232. • TCPIP-Use para Ethernet ou redes sem fio. Para mais informações sobre a configuração de protocolos comuns, clique em **Help** na parte superior da aba.
Description	Digite uma descrição para este canal.

3. Clique **OK**.
   - O canal é adicionado como uma nova linha na tabela.
4. Digite ou selecione as informações para o novo Canal, conforme necessário.
   - Para adicionar ou remover uma coluna, clique com o botão direito na área de cabeçalho da coluna e marque ou desmarque colunas.

Coluna	Descrição
Name	Digite um nome para o canal. O sistema indicará se o nome não é válido, mostrando a célula com o contorno vermelho.
Protocol	Mostra o protocolo que foi selecionado
ProtocolOptions	Configure as opções específicas para este protocolo. **Nota:** As opções de protocolo são dependentes do protocolo selecionado. Selecione o protocolo na como-box na parte superior da página e pressione o botão HELP do seu lado para acessar a documentação do protocolo específico.
Interface	Mostra a interface física usada por este canal.

Coluna	Descrição

Settings	Configure as definições para este canal. Os valores disponíveis dependem da interface que o canal está usando.
	Nota: As configurações aqui devem corresponder às configurações do dispositivo mestre ou escravo com o qual o canal vai conectar.
	• Para a interface serial, normalmente manter os padrões.
	• Para uma interface MultiSerial, digite o número de portas RS-232 para usar no campo Portas.
	• Para a interface TCPIP:
	  • AcceptUnsolicited - Aceitar mensagem não solicitada de escravo.
	  • ListeningPort- Porta TCP, onde o dispositivo escravo está ligado (o padrão é 502).
	  • NodeConnection-Número de pedidos paralelos enviados para cada nó (comunicação assíncrona).
	  • MaxSimultaneousConnections - Máximo número de conexões simultâneas.
	  • ShareNodeSameIP-Um endereço IP único e vários escravos conectados. Por exemplo, RS485/Ethernet ou Terminal Servers.

Timeout	Configure as opções de tempo limite para este canal. Normalmente, mantenha o valor padrão.
IntialState	Selecione o estado inicial para este canal. Habilitado (Enabled) ou não (Disabled).Em qualquer destes casos a task do protocolo será disparada, podendo ser controlada em tempo real (habilitar ou desabilitar). Uma terceira opção Remote, é utilizada para disparo de canais que vão controlar um objeto em outro servidor. A ultima opção Reserved deve ser utilizada para desabilitar completamente um canal (sem o disparo do driver) .
Description	Digite um texto com uma descrição para este canal.
[Outras colunas]	Para definições de outras colunas que estão disponíveis em muitas tabelas, consulte "Definição de colunas comuns" na pagina 68.

5. Continue adicionando quantos canais você precisa.
   - Se necessário, clique com o botão direito em uma linha para cortar, copiar, colar ou excluir uma linha

Configurando Nós

Os nós ou Nodos no Action.NET são os objetos que coincidem com os dispositivos IEDs, ou CLPs com os quais a aplicação deve se comunicar no canal em que estão conectados.

Você também pode importar as configurações de um servidor OPC ou de outra fonte de dados. Consulte "Importando de um Servidor OPC" na pagina 109 e "Importando endereços de PLCs" na pagina 109.

Para configurar os nós:

1. Vá para **Edit > Devices > Nodes**.

2. Digite ou selecione as informações, conforme necessário.

   - Para adicionar ou remover uma coluna, clique com o botão direito na área de cabeçalho da coluna e marque ou desmarque colunas.

Coluna	Descrição
Name	Digite um nome para o nó. O sistema indicará se o nome não é válido, mostrando a célula com o contorno vermelho.
Channel	Selecione o canal para este nó. Para mais informações sobre a configuração de protocolos comuns, clique em **Help** na parte superior da aba.

PrimaryStation	Insira as informações necessárias para acessar o nó principal, com base no protocolo selecionado.
	
	NOTA - As opções de protocolo são dependentes do protocolo selecionado. Selecione o protocolo na combo-box na parte superior da página e pressione o botão **Help** ao lado para acessar a documentação do protocolo.
	
	Para o protocolo Modbus:
	• Para uma interface Serial, o SlaveID é o endereço do dispositivo escravo na rede Modbus. Endereços válidos são 1-247.
	• Para uma interface MultiSerial, selecione o número do ComPort e digite o endereço SlaveID o dispositivo escravo na rede Modbus. Endereços válidos são 1-247.
	• Para uma interface TCPIP:
	  • IP-Identificação do endereço do dispositivo escravo.
	  • Port-TCP, onde o dispositivo escravo está ligado (o padrão é 502).
	  • SlaveID- endereço do device escravo na rede Modbus. Endereços válidos são 1-247.
	Para interfaces OPC:
	• URL do Serviço-Define a localização do servidor OPC.
	  • É necessário configurar as definições de DCOM para acessar um servidor OPC externo. Contate com o suporte para obter assistência.
	• RefreshRate-Taxa de atualização do Server.
	• AllTemsSameGroup - Adiciona todos os itens em um único grupo OPC. Deste modo, apenas uma conexão é criada com o servidor OPC.
	• WaitAfterConnect—Tempo para comunicar depois que a aplicação já está executando.

BackupStation	Insira as informações necessárias para acessar o nó de backup, com base no protocolo selecionado. Quando definido, e uma falha de comunicação ocorre na estação principal, o sistema automaticamente tenta estabelecer comunicação com a estação de Backup.
Description	Descrição Digite uma descrição para este nó.

3. Continue adicionando tantos novos nós quantos você precisar.

Importando de um Servidor OPC

Depois de criar um nó de comunicação OPC, você pode selecionar o nó e clicar em Importar para importar o banco de dados do Servidor OPC para o projeto. O Action.NET cria automaticamente os Tags e os pontos de comunicação (tabela Points).

Depois de ter utilizado a ferramenta de importação, pela primeira vez, o sistema irá salvar as configurações utilizadas, de modo que o botão mostrará a partir de então a palavra SYNC, o que significa que na próxima vez que você usá-lo, será executada uma sincronização, verificando quais endereços já foram importados previamente e quais os novos.

Importando endereços de PLCs

Ao criar os nós de comunicação e pontos de comunicação (Points), você pode importá-los se eles são definidos em outra fonte de dados das seguintes formas:

- Você pode copiar e colar o conteúdo de uma tabela de Excel. As tabelas podem ter diferentes colunas ou ordem, desde que você inclua o título das colunas nas operações de copiar e colar. O sistema irá colocar os dados nas colunas esperadas, mesmo que a ordem seja diferente nas tabelas de origem e de destino.
- Você pode importar os dados de arquivos CSV.
- Para dispositivos Rockwell ControlLogix, você pode importar arquivos de definição L5K.
- Para dados OSIsoft PI ®, existe uma versão Action.NET para compartilhar definições.
- Está disponível também uma API de programação que pode ser usada para escrever scripts preencher as tabelas de Points, até mesmo em tempo de execução, quando é necessário.

Se o dispositivo PLC ou outro IED tem um banco de dados aberto ou arquivo com os endereços disponíveis, e você gostaria de ter uma forte integração para fazer essa configuração e carga de endereços Action.NET, contate o suporte.

Configurando endereços de pontos

A aba **Points** do espaço **Devices** é utilizada para definir os tags que serão adquiridos por este canal e node. Nela são especificados os tipos de pontos (conforme o protocolo) e os identificadores (com os endereços) utilizados pelo protocolo para o mapeamento dos dados dentro dos IEDs.

O número de pontos de dados que você pode configurar está relacionada tanto ao ProductModel configurado para o projeto e sua licença para Action.NET. Para obter informações sobre os modelos de produtos, consulte "Família e Modelo do Produto" na pagina 218.

Para configurar os pontos de dados:

1. Vá para **Edit > Devices > Points**.
   - Você pode copiar e colar as etiquetas da aba Tag> Objetos.
2. Digite ou selecione as informações, conforme necessário.
   - Para adicionar ou remover uma coluna, clique com o botão direito na área de cabeçalho da coluna e marque ou desmarque colunas.

Coluna	Descrição
TagName	Digite um nome de tag ou clique ... para selecionar um tag. Você também pode criar uma nova tag neste ponto.
Node	Selecione o nó para este ponto de dados.
Address	Digite o endereço de registro, com base no PLC e protocolo para este ponto de dados e tag. **NOTA -** As opções de protocolo são dependentes do protocolo selecionado. Selecione o protocolo na caixa de combinação na parte superior da página e pressione o botão **HELP** do seu lado para acessar a documentação do protocolo específico
DataType	Selecione o tipo de dados que você deseja usar. A maioria dos protocolos deve usar a opção nativa. Quando nativo é usado, o protocolo vai tratar automaticamente a conversão de dados. Selecionando um diferentes tipos de dados substitui os padrões. Algumas opções podem não ser aplicáveis a o nó selecionado. Verifique se você sabe os tipos de dados aplicáveis.
Modifiers	Se o PLC usa uma ordem de byte diferente, selecione as opções desejadas. Você pode alterar a posição de bit, byte, Word ou DWORD dos dados que são usados na comunicação.
AccessType	Selecione o tipo de acesso para este ponto. Você pode configurar os tipos de acesso. Veja .em "Configurando Tipos de Acessos" na pagina 111.
Scaling	Se você quiser manipular o valor do tag, de alguma forma, selecione as opções desejadas. Para a opção **Equation** será aplicada quando da leitura dos dados: • Div-O sistema irá dividir o valor recebido com o que você entra aqui. • Add-O sistema irá adicionar a quantidade que você entra aqui como uma compensação para o resultado da divisão. • Para uma operação de escrita, os cálculos são o oposto (multiplica o valor em Div, e em seguida, subtrair o valor Add).
[Outras colunas]	Para definições de outras colunas que estão disponíveis em muitas tabelas, consulte "Definição de colunas comuns" na pagina 68.

3. Continue adicionando tantos pontos quantos você precisa.

Configurando Tipos de Acessos

Na aba de **Tipos de Acesso** (Access Types) são definidos os procedimentos de leitura e escrita, com tempos de amostragem, eventos não solicitados e outras propriedades, que serão utilizados no tratamento de cada um dos tags definidos na aba **Points**.

Você pode configurar os tipos de acesso que controlam ler, escrever e outras configurações quando o aplicativo acessa dados do PLC ou IED..

O Action.NET vem com alguns tipos de acesso predefinidos que você pode usar, ou você pode criar outros próprios, sempre de acordo com as possibilidades existentes no protocolo em questão,

Para configurar os tipos de acesso:

1. Vá para **Edit > Devices > AccessTypes**..
2. Siga um dos seguintes procedimentos:
   - Para editar um tipo de acesso existente, clique duas vezes em um campo.
   - Para criar um novo tipo de acesso, clique em **Create New** .
3. Digite ou selecione as informações, conforme necessário.

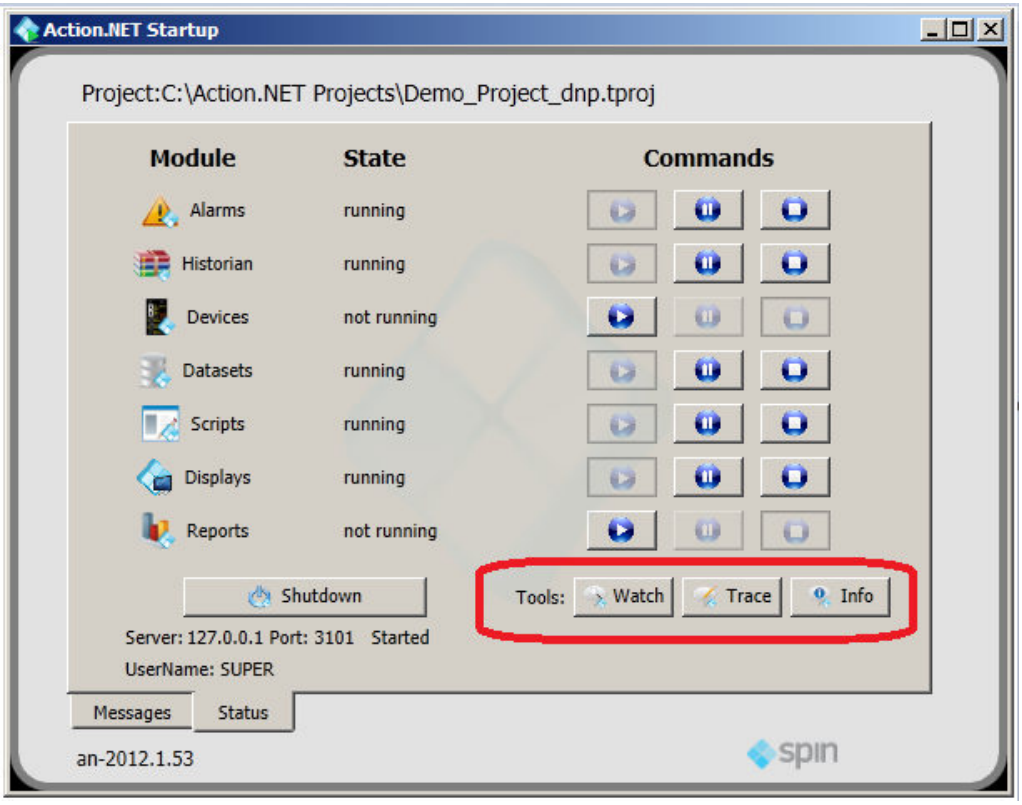
Coluna	Descrição
Name	Digite um nome para este tipo de acesso
Read	
ReadPolling	Selecione a opção se você deseja habilitar a leitura por amostragem
PollingRate	Digite o período de amostragem em milissegundos
ReadTrigger	Digite uma propriedade de objeto para informar ao sistema quando deve ler o valor, por evento.
OnStartup	Quando selecionado, o sistema faz uma leitura na inicialização.
Write	
WriteEnable	Selecione a opção para permitir a gravação de valores para o PLC.
WriteEvent	Selecione o evento que vai disparar a escrita do valor: Pode escolher entre Change, ChangeUp ou ChangeDown ocorrido no valor.
WriteTrigger	Digite uma propriedade de objeto para informar ao sistema quando escrever o valor.
Settings	
AcceptUnsolicited	Quando selecionado, o sistema aceita valores de PLC, mesmo se o tempo de amostragem não tenha expirado.
UseStaticBlocks	

Coluna	Descrição
Description	Digite uma descrição para o tipo de acesso.
[Outras colunas]	Para definições de outras colunas que estão disponíveis em muitas tabelas, consulte "Definição de colunas comuns" na pagina 68.

Ferramentas para diagnose

Depois de iniciada a execução de um projeto voce pode, utilizando a Startup window, selecionar algumas ferramentas de diagnose: As janelas PropertyWatch (Watch), TraceWindow (Trace), e ModuleInformation (Info).

Você também pode iniciar as ferramentas de diagnóstico nas páginas Run-Test e Run-StartUp pressionando o botão esquerdo do mouse sobre os ícones das ferramentas de diagnóstico. Se os menus estão habilitados para as Telas você também pode acessar o menu Ferramentas em tempo de execução.



Info - Informação dos módulos

A janela Info - Module Information contém informações sobre o funcionamento dos módulos. Por exemplo ao escolher um módulo de Device e um canal específico, tem-se várias informações sobre o funcionamento do canal de comunicação.

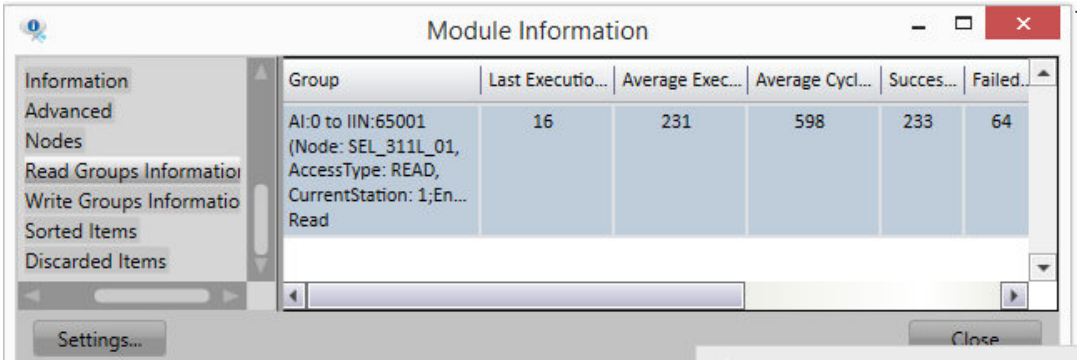
Um ponto muito importante é o Ler Grupos de Informação, pois fornece informações sobre os grupos virtuais de leitura, tempo de execução de cada item, quantidade de leituras e leituras que falharam, e também relatórios sobre o código e data / hora do último erro.

Para a investigação de problemas, os seguintes passos tipicamente devem ser seguidos quando se utiliza o Módulo de Informação:

1. Vá para "Read Groups Information", para verificar os números de sucessos e falhas na comunicação, a fim de identificar rapidamente os blocos de comunicação.
2. Se você tem um erro sistemático em todos os blocos, ou códigos de status com valores negativos, normalmente isso significa que você não pode acessar o dispositivo remoto. Verifique se o endereço do nó está correto.
3. Se você tem um ou outro bloco com erro sistemático, verifique as tags e endereços relacionados com o bloco. Use o TraceWindow com informações de Devices para coletar informações sobre esses erros de comunicação.
4. Para alguns protocolos, como OPC, os itens descartados vão mostrar endereços errados na configuração.

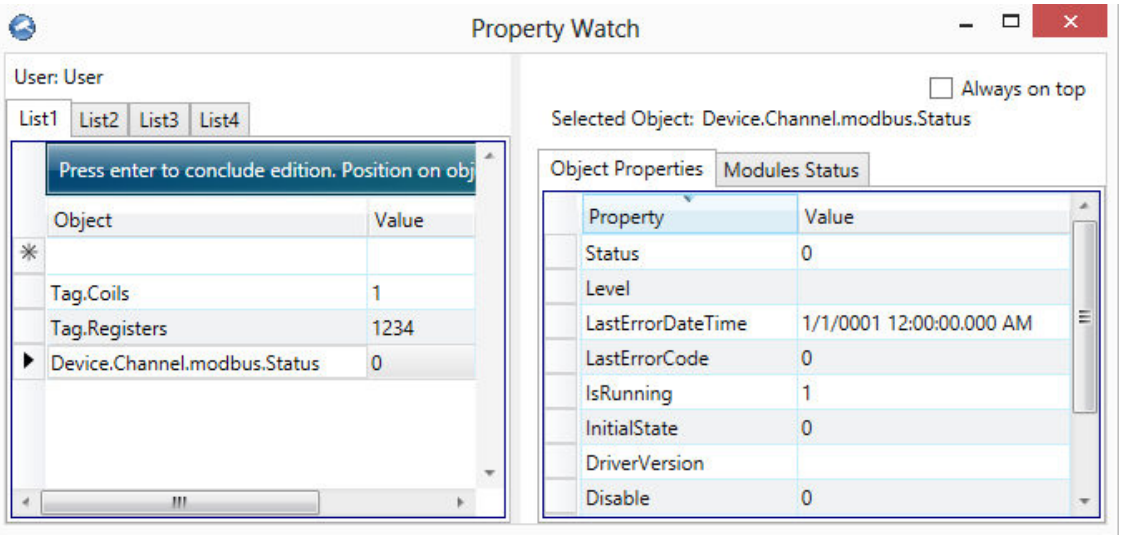
Ao executar o aplicativo no modo teste no Modelo Enterprise, tenha em mente que neste modo, somente são executadas leituras a partir de dispositivos de campo, mesmo que você tenha uma configuração para escrever para o campo.

É muito útil executar o aplicativo com ONLINE CONFIGURATION habilitada, para que não seja preciso parar/iniciar o módulo Device ao modificar a configuração. Você pode modificar os endereços PLC em Points, os AccessTypes e a maior parte da aplicação e ver em tempo real os resultados na execução. Você pode usar o Startup-Window ou o Propertywatch para iniciar e parar apenas um módulo, como os Devices, em vez de reiniciar todo o sistema de execução.



Watch - Tags, Estados e Propriedades

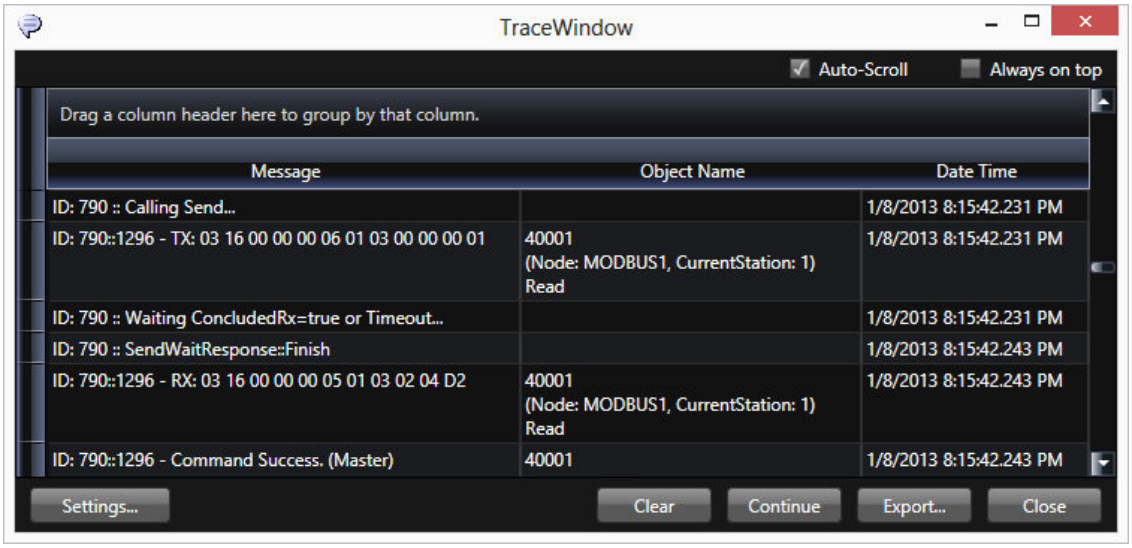
O Property Watch é uma ferramenta de diagnóstico usado para o acesso de atributos em tempo real e propriedades de Tags internas do sistema para leitura ou escrita. Basta digitar o nome da propriedade do objeto na coluna e obtém-se o seu valor na coluna Valor.



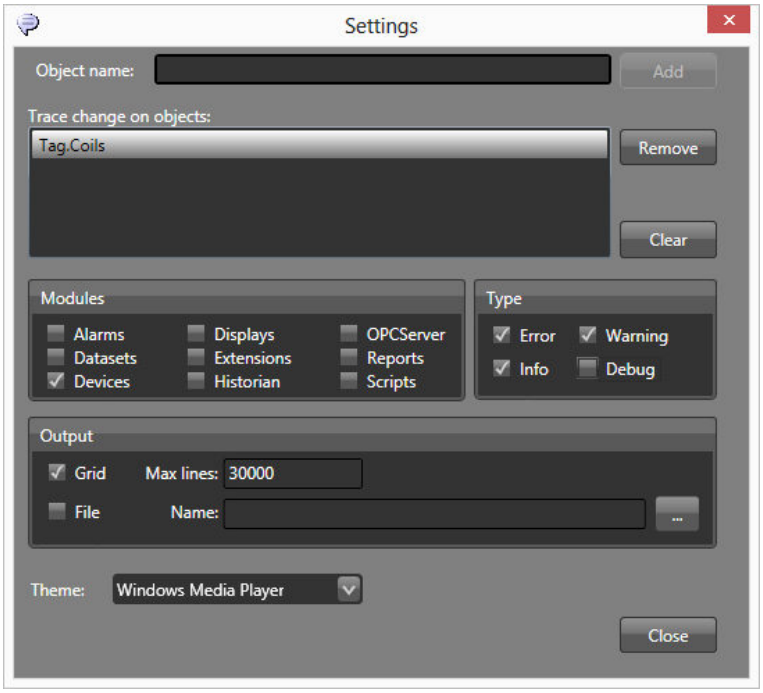
Ex: Tag.Coils, Device.Channel.Modbus.Status. No lado direito, temos propriedades do objeto selecionado.

Janela de Trace

A janela de Trace (rastreamento) é a ferramenta que informa as mensagens do sistema em uma interface do tipo DataGrid (tabela). Ao ativar os dispositivos do módulo no botão Configurações, temos informações sobre o estado de leituras, escritas, não solicitados, quadros TX (envio) e RX (recebido).



Quando se clica no botão **Settings** na aparece uma janela de configuração onde se pode selecionar quais tipos de mensagens e módulos, que serão mostrados no DataGrid de trace. Estes dados que aparecem podem também ser salvos em arquivos para uma análise mais completa. Também é possível configurar um tag em ObjectName clicando no botão **Add**. Ai aparece uma janela com um menu para selecionar objetos para incluir no monitoramento.



Dica: Ao marcar o CheckBox de Devices nas configurações, habilitar apenas o ERROR, INFO e informações de alerta, e não as informações de depuração, caso contrário, você irá criar muitos dados. Para vários dispositivos e protocolos é muito importante usar essa ferramenta, uma vez que o sistema irá apresentar aqui possíveis endereços inválidos da configuração.

Objetos runtime Devices

O namespace **`Device`** é o ponto de entrada para todos os objetos relacionados com o módulo de dispositivos ou Devices.

O objeto **`Device.Channel`** lista todos os canais configurados e suas propriedades em tempo de execução.

O objeto **`Device.Node`** lista todos os nós configurados e suas propriedades em tempo de execução

O objeto **`Device.AccessType`** lista os tipos de acesso definidos e tem opções para executar chamadas síncronas sobre a leitura e a escrita para o dispositivo.

As seguintes propriedades de tags são atualizados com base no módulo de dispositivo:

`tag.tagname.DevicePoint`: endereço no ponto de dispositivos conectados com esta tag

Veja http://www.spinengenharia.com.br/help/an-2014/runtime/index.html para a referência de programação completa em objetos de tempo de execução.

Códigos de Status

A lista a seguir apresenta códigos utilizados para indicar em tempo real o status de um node ou canal.

Estes códigos estão em tempo real em

```
Device.Node.<node name>.Status e em
Device.Channel.<channel name>.Status
```

Lista de códigos:

0 Success
-1 BuildCommandException
-2 ParseCommandUnsolicitedException
-3 ParseReplyException
-4 BuildReplyUnsolicitedException
-5 ChannelException
-6 NodeException
-100 Base Send Error
-101 Base SendAndWait Error
-102 TCP Create Error 1
-103 TCP Create Error 2
-104 TCP Create SocketError
-105 TCP Connect Callback Error
-106 TCP Receive Error
-107 UDP Create Error
-108 UDP Receive Error
-109 Serial Create Error
-110 Serial Receive Error
-111 TCP NotConnected
-112 Start message timeout
-113 Receiving bytes timeout
-114 End message timeout
-115 Connect timeout
-200 ProtocolError
-201 InvalidProtocol
-202 InvalidStation
-203 InvalidCommand
-204 InvalidMsgSequence
-205 InvalidCheckSum
-206 InvalidAddress
-207 InvalidModifiers

Valores positivos são utilizados como status de erros para protocolos específicos.

Alarmes, eventos e Audit Trail

As seguintes seções descrevem como configurar Alarmes e Eventos.

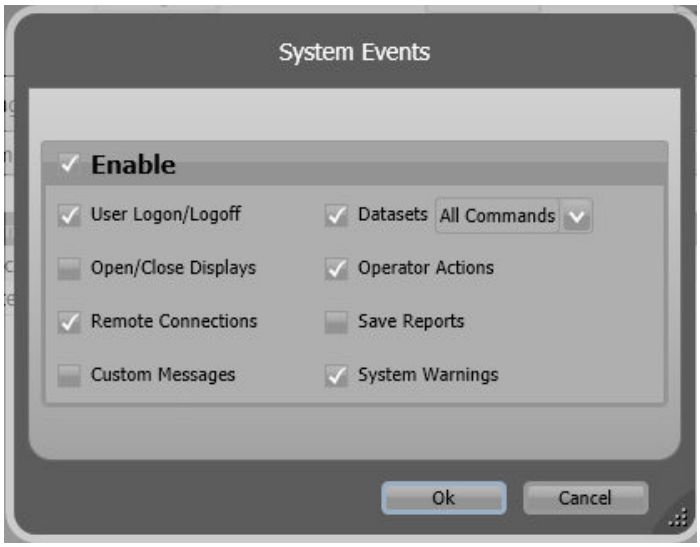
- "Configurando Audit Trail" na pagina 117
- "Configuração das opções de alarme" na pagina 118
- "Configurando grupos de alarme" na pagina 118
- "Configurando Itens de alarme" na pagina 120
- "Visualizando Alarmes e Eventos" na pagina 124
- "Reconhecimento de alarmes" na pagina 124
- "Inscrevendo-se em Notificações" na pagina 124
- "Banco de dados de alarmes" na pagina 125
- "Objeto Runtime Alarm" na pagina 125

Configurando Audit Trail

Você pode selecionar quais eventos devam ser registrados nos arquivos de Audit Trail (registros para rastreamento de auditoria) Esta configuração é válida para os eventos de sistema que você deseja selecionar, abaixo são os do grupo predefinido **SystemEvent**.

Para configurar o Audit Trail:

1. Vá para **Edit> Alarms > Groups** .
2. Clique **Settings** .
   - É mostrada a janela de seleção de **System Events**. Marque os tipos de eventos que deseja sejam registrados.



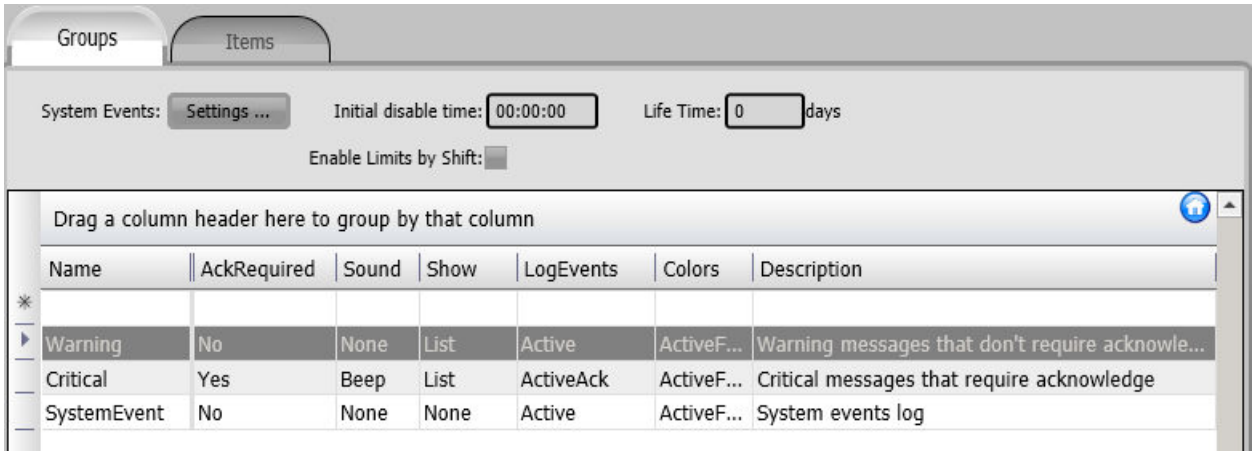
3. Para habilitar o registro de eventos, na tabela da aba Groups coluna **LogEvents**, selecione a opção Ativar
4. Depois de selecionar Ativar, selecione os eventos que você deseja registrar.
5. Clique em **OK**.

Configuração das opções de alarme

Você pode configurar as definições gerais de comportamento para alarmes de eventos que determinam a forma como o sistema trata os alarmes durante a inicialização e por quanto tempo os alarmes e eventos devem permanecer no Historiador de alarmes.

Para configurar as opções de alarme gerais:

1. Vá para **Edit> Alarms > Groups**.
2. Na caixa de texto Initial Disable Time, na parte superior da aba, digite quanto tempo o sistema deve esperar durante a inicialização antes de considerar que um estado de alarme deve ser um alarme.
   - Durante a partida do sistema, os IEDs podem apresentar alarmes, por não ter sido inicializados. Esta configuração informa ao sistema para esperar pelo período de tempo configurado, para deixar que se complete a inicialização, antes de o sistema considerar alarmes.
3. No campo Time Life, digite o tempo que o Historiador de alarmes deve manter os dados de alarme, em dias.



Configurando grupos de alarme

Grupos de alarme permitem que você configure as definições para o que acontece quando ocorre um alarme. Use grupos de alarme para configurar definições de comportamento comuns para uso com

diversos eventos de alarme. As configurações do grupo determinam ações como se uma confirmação é necessária, se um som é reproduzido, o que é registrado, e como são mostrados os alarmes.

O Action.NET vem com alguns grupos de alarmes pré-definidos que você pode usar, ou se preferir você pode criar outros conforme suas necessidades e critérios.

Para criar um novo grupo, defina o nome do grupo em EditAlarmGroups onde o comportamento do item de alarme é especificado.

Os grupos pré-definidos no Action.NET são:

· **Critical** - mensagens críticas que necessitam de reconhecimento.

· **SystemEvent** - Eventos de sistema que podem ser gravados na Audit Trail.

· **Warning** -Mensagens de aviso que não necessitam de reconhecimento.

Para configurar grupos de alarme:

1. Vá para **Edit> Alarms > Groups** .
2. Digite ou selecione as informações, conforme necessário.

Coluna	Descrição
Name	Digite um nome para o grupo de alarme. O sistema indicará se o nome não é válido, mostrando a célula com o contorno vermelho.
AckRequired	Se escolhido Yes, nesta opção, o alarme permanece na lista de alarme até que alguém reconheça o alarme, clicando duas vezes no aplicativo.
Sound	Selecione, entre alguns disponíveis, típicos do MS-Windows, o som que será reproduzido quando ocorre o alarme.
Show	Selecione **List** para que uma mensagem de alarme seja exibida na janela do Objeto de Alarmes na aplicação
LogEvents	Selecione em que condições você deseja que o alarme seja registrado no Historiador de alarmes: • **None**-Alarme não será logado. nunca. • **Active** - Registra o alarme quando ele está ativo. • **ActiveAck** - Registra quando o alarme está ativo e reconhecido. • **ActiveNorm** - Registra quando o alarme retorna ao normal. • **All** - Registra em todas as condições acima.
Colors	Selecione as cores que deseja usar para cada estado, tanto para o texto como para o fundo do texto. Os estados são: • **Active**- Tag está no estado de alarme. • **Normalizado**- Tag estava no estado de alarme, mas já foi para o estado normal e ainda precisa ser reconhecido. • **Acknowledge** - Reconhecido- Tag foi reconhecido, mas ainda está em estado de alarme.
ACKtimeout	Define um tempo limite para o reconhecimento do alarme. Se o alarme não for reconhecido após o tempo especificado, o alarme volta para o estado ativo novamente

Coluna	Descrição
AutoAckTime	Se o alarme não for reconhecido após o tempo especificado, o sistema reconhece o alarme automaticamente
Description	Digite uma descrição deste grupo de alarme.
[Outras colunas]	Para definições de outras colunas que estão disponíveis em muitas tabelas, consulte "Definição de colunas comuns" na pagina 68.

3. Continue adicionando tantos grupos de alarme quantos você precisar.
   - Se necessário, clique com o botão direito em uma linha para cortar, copiar, colar ou excluir a linha da tabela.
   - Consulte as seguintes seções relacionadas com as outras abas de alarme: "Configurando Itens de alarme" na pagina 120.

Configurando Itens de alarme

Ao configurar itens de alarmes, você pode configurar os valores limites específicos que devam gerar um alarme. Cada linha nesta tabela de **Itens de Alarme** refere-se a um alarme. Você pode ter várias linhas para o mesmo tag para definir limites diferentes para faixas ou estados múltiplos de um Tag. Por sua vez cada **Item** refere-se a um Grupo de alarme, para definir o comportamento de visualização e ações durante as transições de estado de alarme (ativo / normal).

Para configurar itens de alarmes:

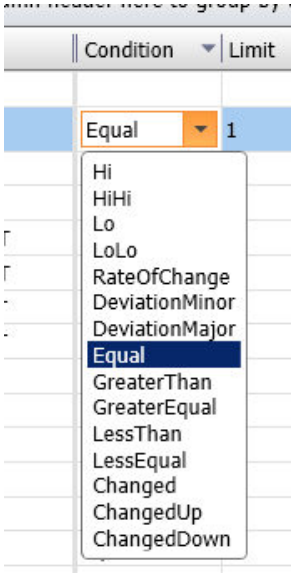
1. Vá para **Edit> Alarms > Itens**.
2. Digite ou selecione as informações, conforme necessário.

Coluna	Descrição
TagName	Digite um nome de tag ou clique ... para selecionar um tag.
Condition	Selecione entre um conjunto de possíveis eventos a condição que deseja usar para a geração deste alarme. Para as condições DeviationMinor ou DeviationMajor, especifique um limite, então use a coluna de valor nominal para definir um valor ou um Tag cujo valor deva ser comparado para obter o desvio atual.
Limit	Digite um valor para o limite de alarme que corresponde à condição que você selecionou.
Group	Selecione um grupo de alarme, entre os já definidos, que será utilizado para controlar o comportamento requerido quando ocorrem as transições de alarme. Veja "Configurando grupos de alarme" na pagina 118

Coluna	Descrição
Priority	Digite um valor de prioridade que controla a posição onde o alarme é exibido no Objeto com a lista de alarmes. Quanto maior o número, maior a prioridade. Você pode usar a mesma prioridade para mais de um evento de alarme. Digite 0 (zero) para os alarmes que devam ir para o fim da lista.
Message	Digite o texto que vai aparecer na lista de alarmes.
[Outras colunas]	Para definições de outras colunas que estão disponíveis em muitas tabelas, consulte "Definição de colunas comuns" na pagina 68.

3. Continue adicionando tantos itens de alarme quantos você precisar.
   - Se necessário, clique com o botão direito em uma linha da tabela para cortar, copiar, colar ou excluir a linha.

EditAlarmsItems.Condition



Esta é a coluna em que se define a condição. Nela podem ser escolhidos os eventos a seguir listados.Veja em cada caso a condição de avaliação que é utilizada para a geração dos eventos de alarmes.

■Hi: Tag > = limite

■HiHi: Tag > = limite (quando reconheceu automaticamente reconhece Hi alarme mesmo Tag)

■Lo: Tag <= limite

■LoLo: Tag <= limite (quando reconheceu automaticamente reconhece alarme Lo a mesma Tag)

■RateOfChange: Taxa de Tag da mudança> = limite

■DeviationMinor: Valor absoluto (tag - Setpoint)> limite (valor nominal definido na coluna do valor nominal)

■DeviationMajor: Valor absoluto (tag - Setpoint)> limite (valor nominal definido na coluna do valor nominal)

■Equal: Tag = limite

■GreaterThan: Tag> limite

■GreaterEqual: Tag> = limite

■LessThan: Tag <limite

■LessEqual >: Tag < = limite

■Changed: valor Tag mudou

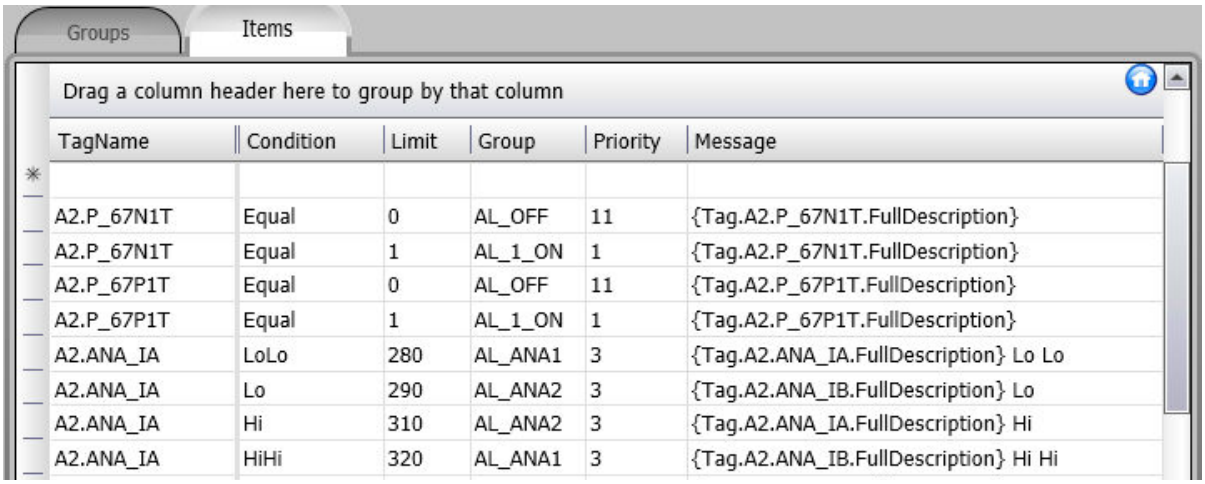
■ChangedUp: valor Tag aumentou

■ChangedDown: valor Tag diminuiu

Múltiplos Itens de Alarme para um Tag

Deve ser observado que um tag pode ter um ou mais registros nesta tabela, em função do comportamento desejado. A figura abaixo mostra exemplos de tags com quatro registros de alarme (variáveis analógicas), dois

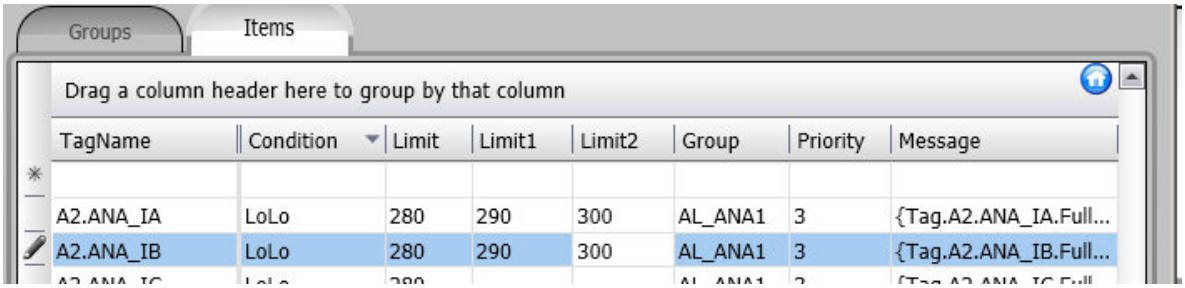
registros (eventos que se deseja dois comportamentos distintos) e um registro de alarme (sinalização de estado de equipamento).



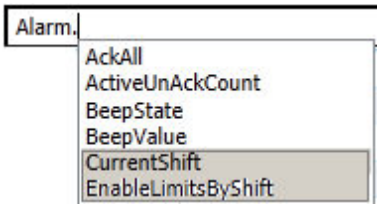
TagName	Condition	Limit	Group	Priority	Message
A2.P_67N1T	Equal	0	AL_OFF	11	{Tag.A2.P_67N1T.FullDescription}
A2.P_67N1T	Equal	1	AL_1_ON	1	{Tag.A2.P_67N1T.FullDescription}
A2.P_67P1T	Equal	0	AL_OFF	11	{Tag.A2.P_67P1T.FullDescription}
A2.P_67P1T	Equal	1	AL_1_ON	1	{Tag.A2.P_67P1T.FullDescription}
A2.ANA_IA	LoLo	280	AL_ANA1	3	{Tag.A2.ANA_IA.FullDescription} Lo Lo
A2.ANA_IA	Lo	290	AL_ANA2	3	{Tag.A2.ANA_IB.FullDescription} Lo
A2.ANA_IA	Hi	310	AL_ANA2	3	{Tag.A2.ANA_IA.FullDescription} Hi
A2.ANA_IA	HiHi	320	AL_ANA1	3	{Tag.A2.ANA_IB.FullDescription} Hi Hi

Habilitar Limites por Turno

A opção "*Habilitar Limites por turno"* é especialmente orientada a sistemas que necessitem implementar a funcionalidade de Níveis de Carga, (Leve, média e Pesada). Quando é escolhida na tabela *Itens*, conforme a figura abaixo, aparecem três colunas para que se especifiquem os limites para cada nível de carga.



TagName	Condition	Limit	Limit1	Limit2	Group	Priority	Message
A2.ANA_IA	LoLo	280	290	300	AL_ANA1	3	{Tag.A2.ANA_IA.Full...
A2.ANA_IB	LoLo	280	290	300	AL_ANA1	3	{Tag.A2.ANA_IB.Full...



O atributo *CurrentShift* informa o turno atual: (0 = leve, 1=médio e 2= pesado)

O atributo *EnableLimitsByShift* informa se a funcionalidade está ou não habilitada, respectivamente. Se habilitada basta criar uma rotina que a cada hora defina o turno (shift): "*Alarm.CurrentShift* = *x*" onde x = 0 / 1 / 2.

Visualizando Alarmes e Eventos

Para visualizar os alarmes e eventos, você pode usar o objeto lista de alarmes disponível para ser colocado nas telas ou relatórios. Consulte a seção "Configurando uma Janela de Alarmes" na pagina 178 para mais informações.

Reconhecimento de alarmes

Existem várias formas para execução do reconhecimento de alarmes:

- Usando o Objeto de visualização de Alarmes nas telas.
- As propriedades do Tags.
- As Propriedades de tempo de execução de grupos de alarme ou de itens de alarme.

Reconhecer todos os alarmes

- Você pode usar a propriedade <Alarm.AckAll> que reconhece todos os alarmes ativos.

Reconhecer um único Alarme ou o Alarme de maior prioridade

- A propriedade <Alarm.PriorityItem.UnAck> permite o reconhecimento do alarme de maior prioridade (configurado no <Edit.Alarms.Items> na coluna "Prioridade"), se houver um reconhecimento de alarme pendente.

Reconhecer um alarme específico

- Para reconhecer um alarme específico usar a propriedade <Alarm.Items.IDxx.Unack>.
- Para verificar o conteúdo das colunas de alarme IDXX, vá para o item Alarme e adicione a coluna ID (clique com o botão direito na tabela e selecione ID).

Inscrevendo-se em Notificações

Para aplicar ações personalizadas usando os scripts dot NET, você pode se inscrever em notificações de alarmes e eventos.

O uso típico é criar um procedimento para enviar um avisos por SMS ou e-mail, executar cálculos personalizados, adicionar mensagens personalizadas de notificação ou alarmes de áudio, alarmes por áudio "text-to-speech", e qualquer tipo de ação personalizada programada usando o Microsoft. NET Framework.

Para se inscrever em eventos de alarmes, é preciso criar um método em qualquer classe do tipo Script Server, com o seguinte protótipo:

```
Notification void (AlarmEventInfo[] info)
```

A cada novo evento de transição de alarme gerado este método será chamado.

Finalmente, você precisa selecionar esse método em **Edit> Alarms> Groups** na Coluna **NotificationMethod** na tabela de Groups.

O nome de método pode variar, o que é importante são os parâmetros esperados do método. A estrutura `AlarmEventInfo` está definida no namespace Alarm na referência as classes de tempo de execução.

Veja em http://www.spinengenharia.com.br/help/an-2014/runtime/index.html .

Banco de dados de alarmes

O banco de dados usado para armazenar os alarmes é definido em **Edit-Datasets-DBs** pelo objeto de conexão de banco de dados com o nome **AlarmHistorian**.

Por default, quando um novo projeto é criado, o **AlarmHistorian** é definido para usar o banco de dados SQL embutido SpinDB.

AVISO - O banco de dados SpinDB deve ser usado para bancos de dados de até 10 GB. Se for prevista a utilização de mais de 10 GB, você deve definir outro gerenciador de SQL para Banco de Dados. do Historiador de Alarmes.

Para definir um outro banco de dados para armazenar o banco de dados do Historiador de Alarmes, você só precisa criar uma nova conexão de banco de dados, como explicado em "Configurando Conexões de Bancos de dados" na pagina 127 e nomeá-la de **AlarmHistorian.**

DICA - O sistema não permite nomes duplicados, de modo que para criar uma conexão DB com o nome AlarmHistorian você deve renomear ou excluir a linha até então existente usando esse nome.

O Módulo de alarme cria automaticamente as tabelas necessárias no banco de dados. Um exemplo de esquema da tabela está disponível ao abrir qualquer arquivo com extensão. TAlarm, criado durante a execução de aplicações usando o banco de dados padrão SpinDB para o registro de alarmes

Objeto Runtime Alarm

O namespace Alarm tem as propriedades do servidor de alarmes.

O objeto Alarm.Group tem a lista de todos os grupos definidos e suas propriedades.

O objeto Alarm.Item tem todos os itens de alarme e suas propriedades.

As seguintes propriedades dos Tags são relacionadas com o módulo de alarme:

- tag.tagname.Hi: configuração e status de execução do alarme HI.

- Os nomes semelhantes para outros tipos de alarme.

Veja em http://www.spinengenharia.com.br/help/an-2014/runtime/index.html , para a referência de programação completa em objetos de tempo de execução.

Datasets e Intercâmbio de dados

As seções a seguir descrevem como configurar os bancos de dados:

- "Configurando Conexões de Bancos de dados" na pagina 127
- "Configurando acesso à Tabelas de BDs" na pagina 133
- "Configurando consultas de banco de dados" na pagina 134
- "Configurando Arquivos para troca de dados" na pagina 135
- "Objetos Runtime Datasets" na pagina 136

Configurando Conexões de Bancos de dados

Por default, o Action.NET usa um gerenciador de banco de dados SQL embutido (SpinDB) para os Historiadores de Tags e de Alarmes. Você pode configurar outros bancos de dados externos, tais como o Microsoft SQL Server, Oracle, ou outros. Se você tem a expectativa de que estes historiadores vão necessitar armazenar mais de 10 GB, você deve considerar desde a implantação inicial o uso de um banco de dados externo.

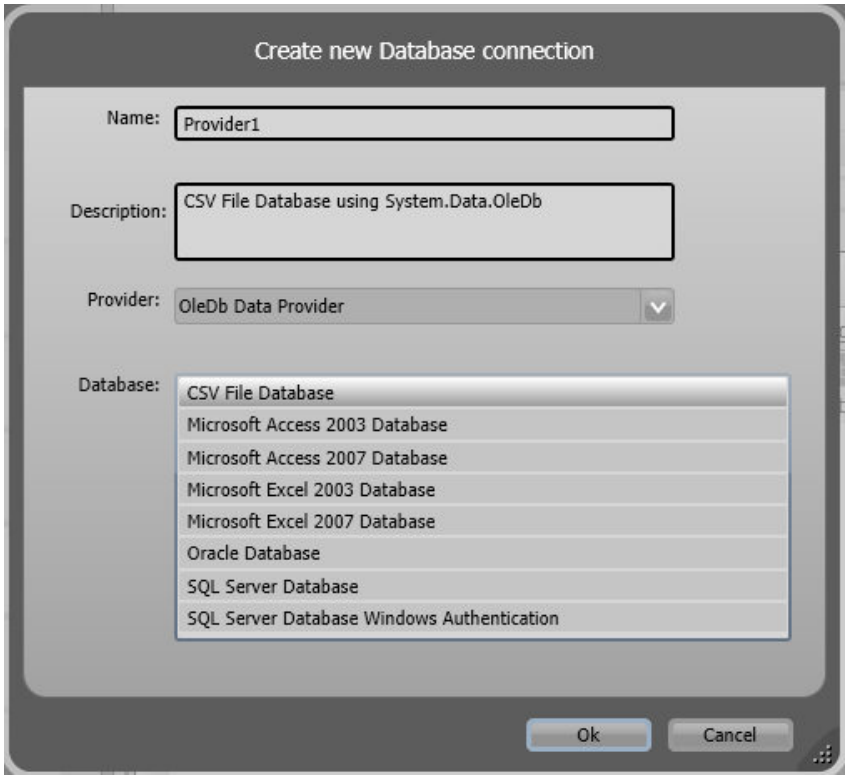
Ao usar o banco de dados integrado, o Action.NET cria automaticamente o arquivo de banco de dados. Ao usar bancos de dados externos, o arquivo do banco de dados deve ser criado pelos utilitários pelo uso do gerenciador escolhido. No entanto, o Action.NET pode criar as tabelas no banco de dados, disponibilizado.

Você pode usar também bancos de dados externos como fonte de dados em sua aplicação para exibir dados como informações do cliente ou de seus produtos ou para configurar sistemas de receitas.

Você precisa estar logado como administrador no Action.NET, para configurar o login e a senha do banco de dados.

Para configurar um banco de dados externo:

1. Vá para **Edit> Datasets > DBs.**
2. Clique em **Create New**.
   - É mostrada a janela de Criar Nova Conexão de banco de dados.
3. Digite ou selecione as informações, conforme necessário



Coluna	Descrição
Name	Digite um nome para a configuração do banco de dados. O sistema indicará se o nome não é válido, mostrando a célula com o contorno vermelho.
Description	Digite uma descrição para o banco de dados.
Provider	Selecione o provedor de banco de dados.
Database	As opções mostradas aqui dependem dos drivers de provedores existentes e instalados na máquina. Selecione o tipo de banco de dados que será usado

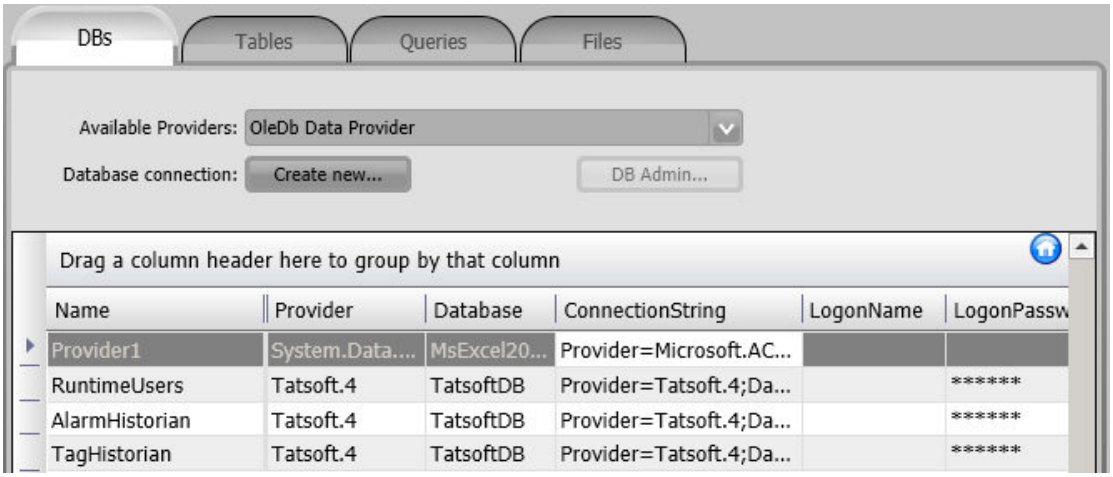
4. Clique **OK**.
   - O banco de dados é adicionado como uma nova linha na tabela.
5. Digite ou selecione as informações, na nova linha da tabela, conforme necessário, para concluir a configuração do banco de dados

Coluna	Descrição
Name	Digite um nome para a configuração do banco de dados. O sistema indicará se o nome não é válido, mostrando a célula com o contorno vermelho.
Provider	Selecione o provedor de banco de dados.

Coluna	Descrição
Database	As opções mostradas aqui dependem dos drivers de provedores existentes e instalados na máquina. Selecione o tipo de banco de dados que será usado
ConnectionString	Digite as informações necessárias para se conectar ao banco de dados.
LogonName	Digite um nome de login válido para o banco de dados.
LogonPassword	Digite a senha que corresponde ao LogonName
Description	Digite uma descrição para o banco de dados.
[Outras colunas]	Para definições de outras colunas que estão disponíveis em muitas tabelas, consulte "Definição de colunas comuns" na pagina 68.

6. Se você estiver usando o novo banco de dados externo para o historiador de tags ou de alarmes, mude o nome dos bancos de dados existentes TagHistorian or AlarmHistorian para outros, e em seguida, mude o nome da nova configuração de banco de dados para TagHistorian ou AlarmHistorian.
   - Para os bancos de dados de historiador de tags ou de alarmes, a configuração dos nomes dos banco de dados nesta aba devem ser TagHistorian ou AlarmHistorian. Você não pode ter dois bancos de dados com o mesmo nome.
7. Continue adicionando tantos bancos de dados quantos que você precisa.
   - As bases de dados que você criar estarão disponíveis para uso na aba **Tags> Historian**, na confecção de telas e para os scripts.
   - Se necessário, clique com o botão direito em uma linha para cortar, copiar, colar ou excluir a linha.

A figura apresenta o espaço de trabalho **Datasets**

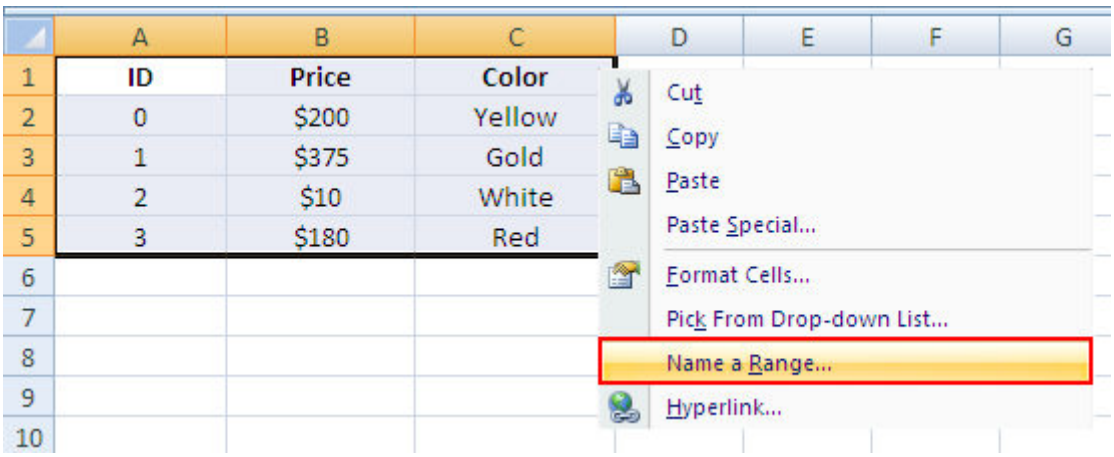


Accessando o Microsoft Excel

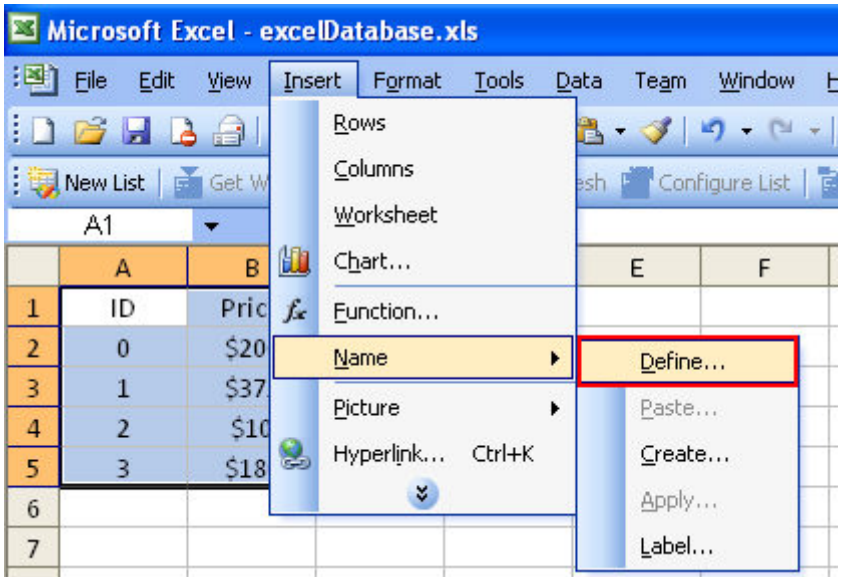
Para conectar-se à bancos de dados do Excel pode-se utilizar um driver ODBC, um DSN ODBC, ou OleDB. As seções a seguir apresentam detalhadamente os passos para configurar cada um destes métodos de conexão.

Criando as bases de dados ODBC

1. Selecione, e atribua um Nome para um intervalo de linhas e colunas na planilha. Isso permitirá que o software possa ler as informações como uma tabela.
2. Escolha um dos seguintes processos de nomeação para a sua versão do Microsoft Excel.
   - Para o Microsoft Office 2007:
     - Botão direito do mouse na área selecionada e escolha "Nome de um Range".



   - Para o Microsoft Office 2003:
     - No Microsoft Excel vá para "Insert > Name > Define".

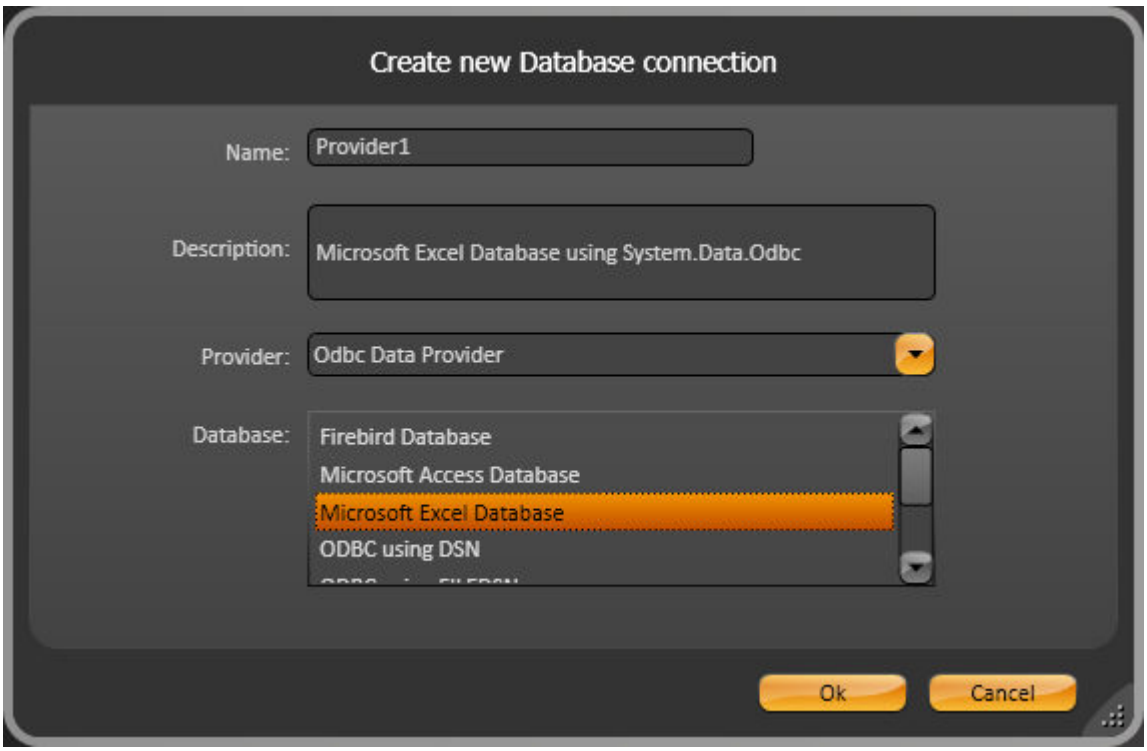


3. Nomeie a seleção (por exemplo, "itensTable"). O arquivo Excel está agora pronto para o uso.

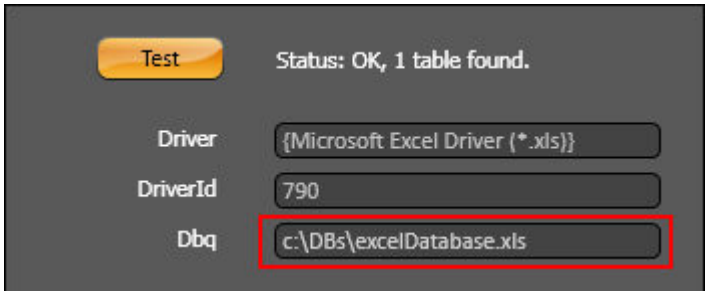
Métodos para se comunicar usando ODBC

Usando ODBC Microsoft Excel Driver:

1. No espaço de trabalho **Datasets** escolha a aba "DBs" e crie um novo Provider clicando em **Create New**
2. Selecione "Odbc Data Provider" no campo de "Provider"
3. No campo "Database" escolha "Microsoft Excel Database".
4. Clique Ok.

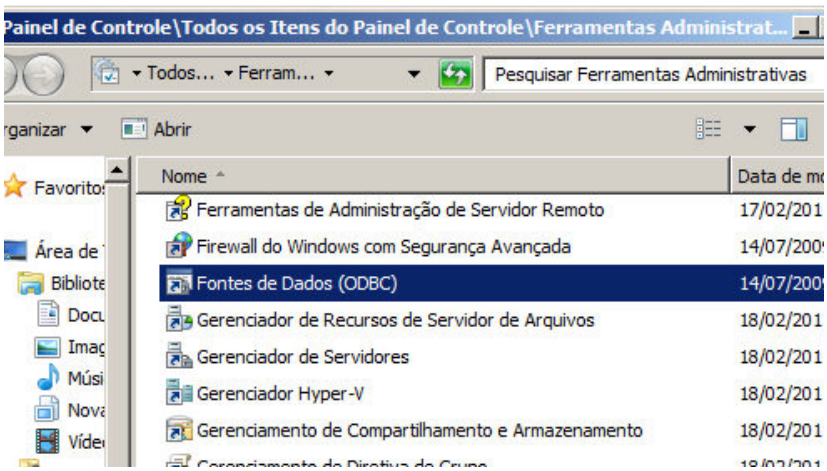


5. Uma nova linha será criada na tabela, clique a coluna "ConnectionString".
6. Aparece uma janela: nela entre com o caminho e o nome do arquivo no campo "Dbq" .
7. Opcionalmente clique o botão "Test" para assegurar-se que a conexão está OK.

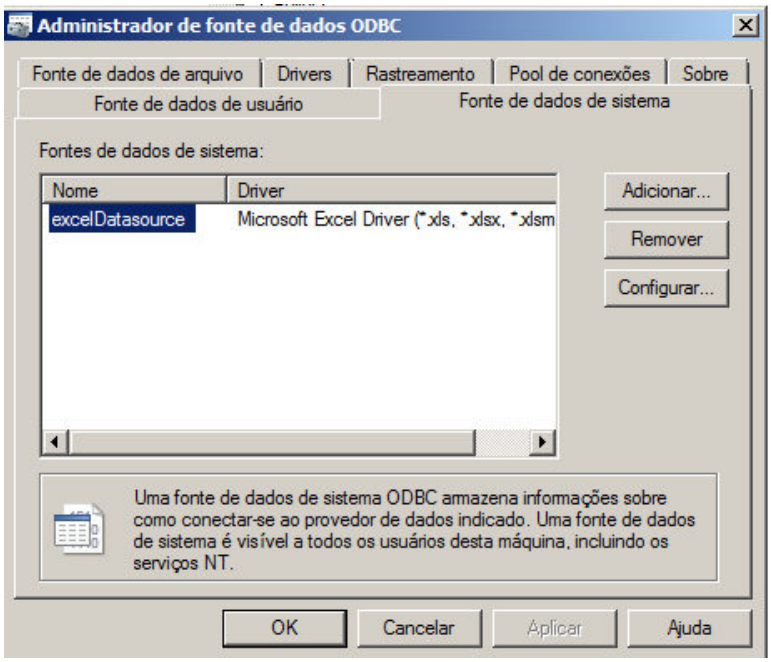


Usando ODBC com um DSN:

1. No MS-Windos vá até “Painel de Controle” e selecione "Ferramentas Administrativas".
2. Faça um duplo clique sobre "Data Sources (ODBC)".



3. Na janela “Administrador de Fonte de Dados ODBC” clique em Adicionar. Aparecerá outra janela solicitando para selecionar um “driver”,
4. Selecione o "Microsoft Excel Driver (*.xls)". Clique em Concluir.
5. Clique Selecione Pasta de Trabalho e selecione o nome do arquivo Excel criado anteriormente.
6. Atribua um nome para a fonte de dados . exemplo:. "excelDatasource".
7. Se pretender fazer acessos de escrita desmarque a opção "ReadOnly".

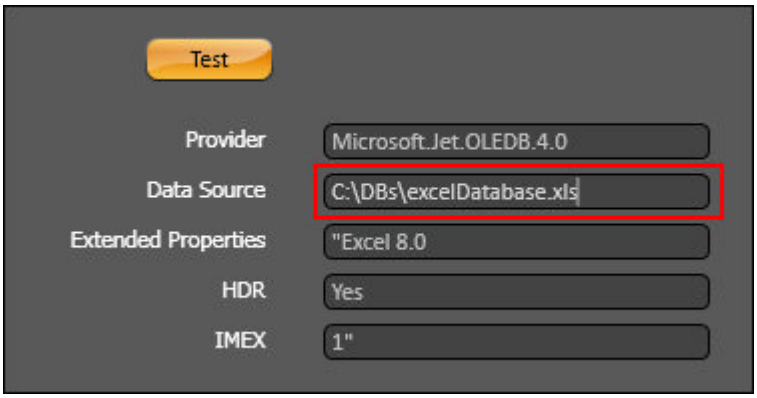


8. No espaço de trabalho **Datasets** escolha a aba DBs e crie um novo Provider clicando o botão **Create New.**

9. Na lista de opções "Odbc Data Provider" escolha "ODBC using DSN" e clique Ok.
10. Clique a coluna ConnectionString da nova linha criada na tabela de conexões e entre com DSN no campo DSN.

Utilizando OLEDB

1. No espaço de trabalho **Datasets** escolha a aba "DBs".
2. Selecione "OleDb data provider" e crie uma nova Conexão clicando em **Create New**
3. Selecione "Microsoft Excel Database" e então clique Ok.
4. Clique na coluna ConnectionString da nova linha criada, e entre com o caminho e o nome do arquivo Excel (.xls) no campo "DataSource"



Configurando acesso à Tabelas de BDs

Se você estiver usando um banco de dados externo como uma fonte de dados em sua aplicação, você pode especificar diretamente qual a tabela que vai utilizar a partir deste banco de dados.

Para configurar as tabelas de banco de dados:

1. Vá para **Edit> Datasets >Tables**.
2. Digite ou selecione as informações, conforme necessário.

Coluna	Descrição
Name	Digite um nome para a configuração da Tabela. O sistema permite que você saiba se o nome não é válido.
DB	Selecione a configuração de banco de dados
TableName	Selecione o nome da tabela.
WhereCondition	Especifique os parâmetros para filtrar os dados usando a sintaxe SQL.
Access	Selecione as permissões de acesso para a tabela.
Mapping	Clique ... para selecionar os Tags que você deseja preencher com os dados da primeira linha da tabela com dados de colunas específicas
Description	Digite uma descrição para a configuração da tabela.
[Outras colunas]	Para definições de outras colunas que estão disponíveis em muitas tabelas, consulte "Definição de colunas comuns" na pagina 68.

3. Continue adicionando tantas configurações de tabelas quantas você precisar.

Ler e escrever o conteúdo da tabela

O acesso, em tempo de execução, para o conteúdo da tabela é executado automaticamente quando a tabela é mapeada para um Objeto DataGrid, colocado em uma tela ou Relatório. Consulte "Configurando uma Janela DataGrid" na pagina 181 .

Você também pode obter o conteúdo da tabela ou executar operações em tabelas, usando as propriedades de tempo de execução do objeto Dataset.Table, consulte http://www.spinengenharia.com.br/help/an-2014/runtime/index.html .

Configurando consultas de banco de dados

Você pode configurar consultas para executar funções mais avançadas com instruções SQL para trabalhar com dados de bancos de dados externos.

Para configurar consultas à banco de dados:

1. Vá para **Edit > Datasets** > **Queries**.
2. Digite ou selecione as informações, conforme necessário.

Coluna	Descrição
Name	Digite um nome para a consulta. O sistema permite que você saiba se o nome não é válido.
DB	Selecione a configuração de banco de dados. da tabela DBs
SqlStatement	Digite a consulta usando a sintaxe SQL.

Coluna	Descrição
Mapping	Clique para selecionar os Tags que você deseja preencher com os dados da consulta com os dados de colunas específicas.
Description	Digite uma descrição para a configuração da tabela.
[Outras colunas]	Para definições de outras colunas que estão disponíveis em muitas tabelas, consulte "Definição de colunas comuns" na pagina 68.

3. Continue adicionando tantas consultas quanto você precisa.

Obtendo o conteúdo da consulta

O acesso, em tempo de execução, ao conteúdo da tabela é executado automaticamente quando a consulta é mapeada para um Objeto DataGrid, colocado em uma Tela ou Relatório. Consulte "Configurando uma Janela DataGrid" na pagina 181

Você também pode obter o conteúdo da consulta ou executar operações de consulta à tabelas, usando as propriedades de tempo de execução do objeto Dataset.Table, consulte http://www.spinengenharia.com.br/help/an-2014/runtime/index.html .

Configurando Arquivos para troca de dados

Você pode configurar arquivos para recuperar dados de um local qualquer de uma rede. acessível

Para configurar os arquivos de banco de dados:

1. Vá para Edit> Datasets > Files.
2. Digite ou selecione as informações, conforme necessário.

Coluna	Descrição
Name	Digite um nome para o arquivo de configuração. O sistema permite que você saiba se o nome não é válido.
FileName	Digite o caminho completo para o arquivo.
FileType	Selecione o tipo de arquivo.
Objects	Clique ... para selecionar os Tags que você deseja preencher com dados do arquivo com os dados de colunas específicas.
Description	Digite uma descrição para o arquivo de configuração..
[Outras colunas]	Para definições de outras colunas que estão disponíveis em muitas tabelas, consulte "Definição de colunas comuns" na pagina 68.

3. Continue adicionando tantas configurações de arquivos quantas você precisar.

Objetos Runtime Datasets

O namespace Dataset é o ponto de entrada para todos os objetos relacionados com o módulo Datasets.

O objeto Dataset.DB. contém a lista de todos os bancos de dados e conexões configuradas e suas propriedades em tempo de execução.

O objeto Dataset.Table lista todas as tabelas configuradas e suas propriedades em tempo de execução.

O objeto Dataset.Query lista todas as consultas definidas e suas propriedades em tempo de execução.

O objeto Dataset.File lista os arquivos de troca de dados definidos e suas propriedades em tempo de execução.

Veja em http://www.spinengenharia.com.br/help/an-2014/runtime/index.html .para a referência de programação completa em objetos de tempo de execução.

Scripts e .NET Framework

Como na maioria dos softwares do tipo SCADA, também está disponível no *Action.NET,* a funcionalidade de permitir ao usuário, durante a criação de um projeto de supervisão, de escrever programas de computador ou *Scripts*. Estes códigos de programação, que serão embutidos no sistema, possibilitam particularizar ou criar comportamentos e ações sobre os objetos (tags, telas, relatórios) não existentes de forma nativa no sistema.

As seções a seguir descrevem como criar scripts:

- "Configurar Tarefas" na pagina 137
- "Configurando Classes" na pagina 139
- "Trabalhando com o Editor de Código" na pagina 140

NOTA - Quando você estiver usando o editor de código, o sistema constantemente compila o código em segundo plano. Você também pode "Build" (construir) todo o projeto, conforme necessário. Para mais informações, consulte "Executando a Aplicação" na pagina 195.

Duas bibliotecas dot NET estão disponíveis para uso comAction.NET:

- Para o Script interno estão disponíveis métodos embutidos que você pode chamar usando TK. <methodName>. Para mais informações sobre esses métodos, consulte o manual Toolkits.html.
- Se você usar o Microsoft Visual Studio, a biblioteca está disponível para uso com o Action.NET. Para mais informações sobre esta biblioteca, consulte a Toolkits.html.

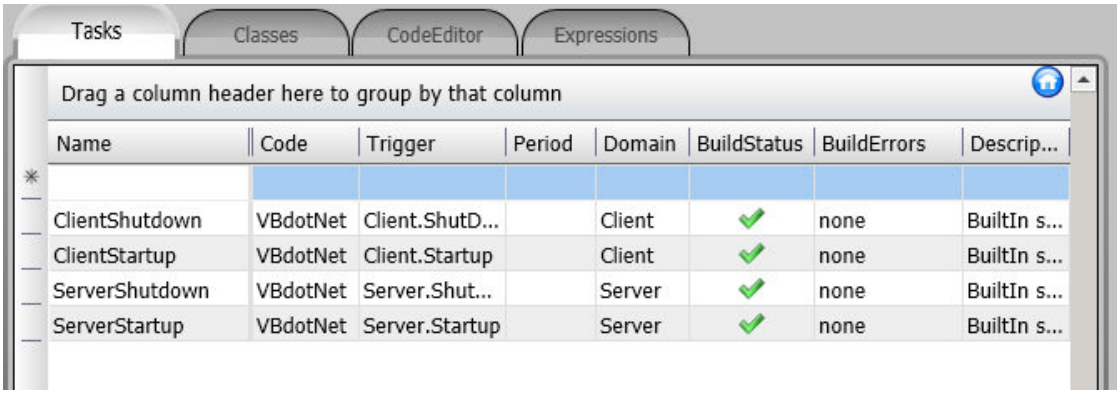
Configurar Tarefas

As Tarefas, ou TASKS são códigos de programas, escrito em VB.Net ou CSharp, que executam a partir de eventos de processo, quando ocorre um evento (trigger) ou periodicamente a cada intervalo de tempo pre definido na configuração

O Action.NET vem com as seguintes Tasks internas, pré definidas:

- **ServerStartup** - Executada quando o projeto começa a ser executado. É executada no servidor (computador que executa TServer.exe).
- **ServerShutdown** - Executada quando o projeto é encerrado. É executada no servidor.
- **ClientStartup** - Executada em cada cliente quando o TVisualizer.exe (módulo DISPLAYS) começa a funcionar.
- **ClientShutdown** - Executada em cada cliente quando o módulo DISPLAYS é fechado.

A figura a seguir mostra a aba TASKS do espaço de trabalho do menu SCRIPTS.



Para configurar tarefas:

1. Vá para **Edit> Scripts> Tasks**.
2. Digite ou selecione as informações, conforme necessário.

Coluna	Descrição
Name	Digite um nome para a tarefa. O sistema permite que você saiba se o nome não é válido.
Code	Somente leitura. Por padrão a linguagem dot Net selecionada quando você criou o projeto. Na guia Editor de código, você pode mudar a linguagem do código. Para alterar o padrão de projeto, vá para **Info > Settings**.
Trigger	Digite ou selecione o evento (tag ou objeto) que será usado para desencadear a execução da tarefa. A tarefa é executada quando o valor do objeto muda.
Period	Digite a periodicidade de execução da tarefa. O campo tem o formato hh:mm:ss
Domain	Selecione o local onde o script é executado: • Cliente-Tarefa executada em cada sistema cliente. Estas são tarefas que se aplicam localmente (no computador do usuário), por exemplo, geração de relatórios. • Servidor-Tarefa executada no servidor do sistema. Estas são tarefas que se aplicam à toda a aplicação, ou seja, a nível global
InitialState	Selecione o estado inicial das tarefas: • Enabled- Tarefa está pronta para ser executado desde o inicio. • Disabled -Tarefa não está pronta para ser executado. Permite ativar a tarefa em condições específicas.
BuildStatus	Somente leitura. Situação do código da tarefa de acordo com o processo de compilação contínua. • Marca Verde - Tarefa compilada sem erros. • Marca em X Vermelha - A tarefa tem erros ou Warnings. Com um duplo clique o sistema vai mostrar o código no ponto em que está o Warning ou erro. Erros impediram a execução da Task específica. O restante da aplicação será executada normalmente.

Coluna	Descrição
BuildErrors	Somente leitura. Mostrar erros encontrados desde o ultimo “Build”
Description	Digite uma descrição para a Task.

3. Após ter completada a entrada de dados na linha descrevendo a TASK digite o código para a Task.
4. Um duplo clique sobre a linha leva-o a acessar a área de código da tarefa na aba **Code Editor**. Veja em “Trabalhando com o Editor de Código” na pagina 140.
5. Ao final da alteração de código clique em **Save** na parte superior do espaço de trabalho.

Configurando Classes

Classes permitem que você crie um repositório de bibliotecas de classes, métodos e funções que você pode usar em toda a aplicação. Você pode chamar estas classe de dentro de Tasks, de outras classes, e de Telas (Code Behind).

O Action.NET vem com as seguintes classes internas:

- Biblioteca ServerMain-Métodos disponíveis para todas as tarefas do servidor.
- Biblioteca ClientMain-Métodos disponíveis para todos os clientes.

Para configurar classes:

1. Vá para **Edit> Scripts> Classes**.
2. Digite ou selecione as informações, conforme necessário.

Column	Descrição
Name	Digite um nome para a classe. O sistema indicará se o nome não é válido mostrando em vermelho o contorno do campo.
Code	Somente leitura. Por padrão a linguagem dot Net selecionada quando você criou o projeto. Na guia Editor de código, você pode mudar a linguagem do código. Para alterar o padrão de projeto, vá para **Info > Settings**.
Domain	Selecione o local onde a classe é executada: • Classe Cliente - Executada em cada sistema cliente. Estas são classes que se aplicam localmente (no computador do usuário), por exemplo, geração de relatórios. • Classe Servidor-Executada no servidor do sistema. Estas são classes que se aplicam à toda a aplicação, ou seja, a nível global

Column	Descrição
BuildStatus	Somente leitura. Situação do código da classe de acordo com o processo de compilação contínua. • Marca Verde - classe compilada sem erros. • Marca em X Vermelha - A classe tem erros ou Warnings. Com um duplo clique o sistema vai mostrar o código no ponto em que está o Warning ou erro. Erros impediram a execução da classe específica. O restante da aplicação será executada normalmente.
BuildErrors	Somente leitura. Exibe todos os erros encontrados durante a última compilação
Description	Digite uma descrição da classe

3. Digite o código para a classe.
   - Dê um duplo clique na linha para acessar a aba **Code Editor** (Editor de código). Consulte Trabalhando com o Editor de Código.
4. Ao final clique em **Salvar,** na barra superior do espaço de trabalho.

Trabalhando com o Editor de Código

Você pode escrever código tanto no VB.Net como em CSharp. Você também pode alternar entre os dois. Se você mudar a sua opção de seleção de linguagem no editor de código, o sistema converte automaticamente o código existente para a linguagem selecionada.

Se você precisa criar referências (References) à suas próprias partes de código (assemblies), você pode usar as facilidades disponíveis em **Run > Build > References.**

Por ser uma aplicação totalmente compatível.com o dot NET, você pode encontrar no meracdo ou na internet, trechos de código fonte, para uso gratuito, inclusive. produtos e bibliotecas dot NET, além de suas próprias bibliotecas que você pode usar imediatamente.

O Action.NET expõe para o programador de Scripts muitas bibliotecas do dot NET que podem ser utilizadas na aplicação. Existem alguns métodos que são muito frequentemente necessárias, tais como a conversão de tipo, ou cópia de Tags para DataTables o para objetos dot NET, ou para configurar dinamicamente parâmetros de comunicação, que estão incluídos em uma biblioteca kit de ferramentas própria. Para usar esses métodos, você só precisa colocar TK. no editor de código, pré fixando o método.

Exemplo:

```
double x = TK.ConvertTo <double> ("123");
```

Os métodos disponíveis estão descritos na Toolkits.html.

Para editar o código:

1. Vá para **Edit> Scripts> CodeEditor** .
2. Na lista drop-down na barra de ferramentas principal, selecione a Tarefa ou Classe que você deseja editar.
   - Para criar uma nova Task ou Classe, consulte "Configurar Tarefas" na pagina 137 ou "Configurando Classes" na pagina 139.

- Caso seja necessário, a partir da barra de ferramentas do editor de código, selecione uma linguagem de código fonte diferente.

3. Clique no ícone de **Salvar** na barra de ferramentas na parte superior do espaço de trabalho.

Configurando Expressões

A ultima aba no espaço de trabalho **Scripts** é utilizada para a criação de **Expressões** que também funcionam como uma linha simples de código que será executada em determinadas condições, fornecendo um resultado.

Além da tabela que pode ser atualizada por **Edit> Script> Expressions**, em vários outros lugares no espaço de Engenharia do Action.NET, você pode digitar e usar expressões, utilizando a mesma sintaxe descrita aqui, para obter um valor resultado da expressão ou para configurar uma condição.

Expressions são meramente codificações na sintaxe VB.Net, como expressões aritméticas ou chamadas para métodos de classes definidas ou acessíveis no projeto. A facilidade Intellisense está orientada apenas para Tags e objetos de aplicação, não considerando a linguagem, mas as expressões são compiladas usando o compilador padrão VB.Net, então o que você puder escrever sem erros de sintaxe no editor de código, também poderá escrever para o código das Expressões.

O Action.NET implementa algumas substituições automatizadas, como == para =, então a sintaxe de uma expressão é muito próxima de uma declaração em C#, mas sem a necessidade de adicionar o ";" no final.

Desta forma, tanto os programadores VB.NET como os de C # são capazes de usar os campos de expressão, sem diferenças.

Em expressões, você não precisa colocar @ antes dos nomes de Tag. Você precisa do @ no editor de código para diferenciar os Tags do projeto das variáveis dot NET. No entanto, como as expressões não têm Variáveis locais dot NET, então pode-se usas os objetos de projeto diretamente.

Para operadores aritméticos, você deve usar os operadores padrão conforme descrito na documentação do dot NET.

Para permitir a avaliação de uma única linha, a linguagem dot NET dispõe do comando IIF, que atualmente é utilizado apenas com IF. O comando IF tem três parâmetros.

Por exemplo:

```
IF (A, B, C)
```

- O primeiro parâmetro é uma condição.
- Esse método retornará B se a condição A é verdadeira;
- E ele retorna C, se a condição for falsa.

Exemplo: Se (tag.A = 1, "Verdadeiro", "Falso") irá retornar uma string de acordo com o valor do tag.A.

Neste método dotNET, todos os três parâmetros são avaliados independentemente da condição. Por exemplo, se você tem

```
IF (tag.A = 1, script.class.client.Func1(),script.class.client.Func2())
```

Ambos Func1 e Func2 vão sempre ser executados. Apenas o valor de retorno será diferente com base no valor de tag.A.

O método IF ou IIF precisa avaliar os parâmetros antes de chamar o método. Há muitas situações em que você deseja executar apenas a função de acordo com o valor.

Para esse cenário o Action.NET tem um método chamado TIF, que tem esse comportamento.

Então, você usa a expressão:

```
TIF (tag.A = 1,script.class.client.Func1(),script.class.client.Func2())
```

Apenas a Func1 () ou a Func2 () vai ser executada, de acordo com o valor de Tag.A

O método TIF é definido na biblioteca de classes que é automaticamente exposta a expressões que estão na biblioteca de funções Toolkit.

Para cálculos mais complexos, você pode chamar uma classe que você deve antes criar na aba Classes. Veja em "Configurando Classes" na pagina 139.

Para configurar expressões:

1. Vá para **Edit> Scripts> Expressions** .
2. Digite ou selecione as informações, conforme necessário.

Coluna	Descrição
Object	Selecione uma tag ou objeto existente.
Expression	Digite a expressão. A expressão pode ser uma expressão matemática básica, usar uma classe, ou ser uma expressão condicional.
Domain	Selecione onde a expressão é executada: • **Cliente** - Expression será executada em cada sistema cliente. Estas são expressões que se aplicam localmente (no computador do usuário), por exemplo, geração de relatórios. • **Servidor** - Expression será executada no servidor do sistema. Estas são expressões que se aplicam em toda a aplicação, ou seja, em nível global.
Execution	Selecione quando a expressão é executada: • OnChange - A expressão é executada quando o valor de qualquer tag na expressão muda. • TriggerOrPeriod - A expressão é executada quando ocorre o evento de disparo configurado ou quando o é expirado o intervalo de tempo definido no período. • ChangeOrStartup - A expressão é executada quando o valor de qualquer tag na expressão muda ou na partida do aplicativo.
Trigger	Digite ou selecione um Tag ou objeto que será usado para disparar a execução da expressão. O disparo da execução ocorrerá quando o valor do tag ou objeto se alterar.
DisableCondition	Digite ou selecione um Tag ou objeto que será usado para desabilitar a execução da expressão

Coluna	Descrição
BuildStatus	Somente leitura. Situação do código atualizado após clicar em **Verify.** • Marca Verde - Expressão compilada sem erros. • Marca em X Vermelha - A Expressão contém erros.
BuildErrors	Somente leitura. Exibe todos os erros encontrados durante a última compilação

3. Clique em **Verify** para verificar a validade da expressão.

Usando o depurador dot NET

Criando Informação de depuração

O Action.NET tem um depurador dot NET integrado. Para usá-lo, é necessário que o computador local tenha arquivos de cache com os arquivos necessários para executar o depurador.

As etapas para ativar o depurador são:

1. Ativar o Debug Informações em **Run> Build> Messages**
2. Se necessário, salve novamente o código fonte que você deseja depurar, deste modo as informações de depuração serão criadas. Esta etapa só é necessária na primeira vez que abrir o projeto no computador. Depois disso, a compilação de "background" irá manter a geração das informações de depuração, à medida que se altera o código, para permitir o uso de pontos de interrupção e execução passo a passo.

Anexando o depurador dot NET

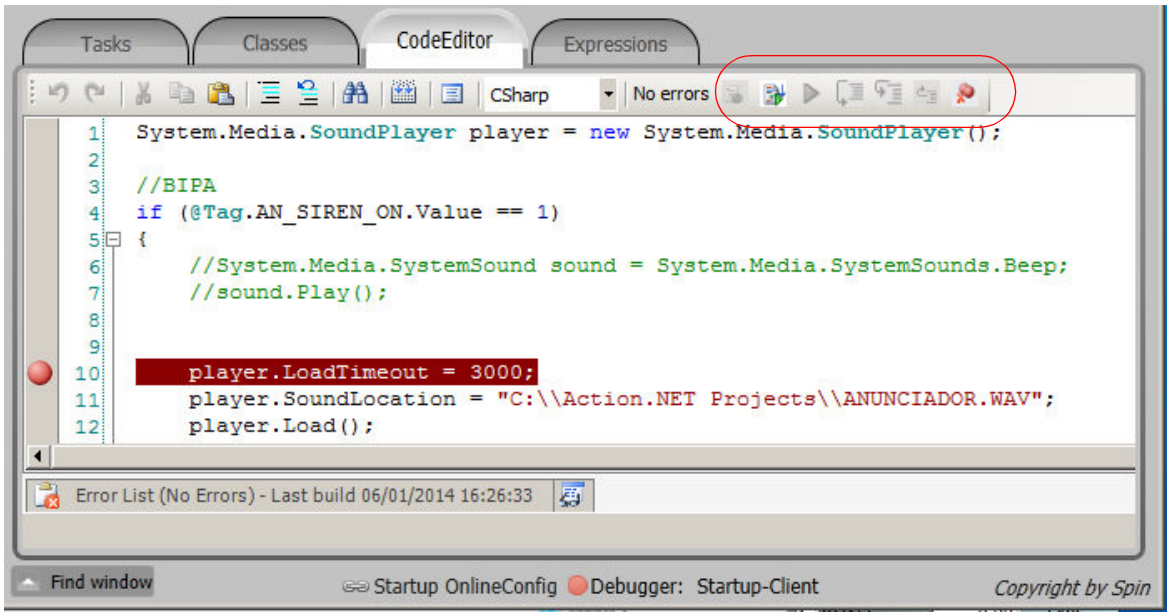
De modo a se ter uma sessão de depuração dot NET, o ambiente de Engenharia deve ser conectado ao ambiente do runtime e o depurador dot NET anexado ao processo do servidor ou processo do cliente.

Siga estes passos.

1. Durante a execução do projeto, seja em **Run>Test** ou **Run>Startup** habilitar a caixa de seleção de conexão.
2. Se o projeto já estiver em execução, você pode ir para **Run>Test** ou **Run>Startup**, de acordo com o tipo de execução que você deseja anexar, e conectar-se ao sistema de execução, pressionando o botão de conexão nessas páginas.
3. Abra qualquer script que tem informações de depuração e pressione o botão **Anexar. depurador NET**. Uma mensagem na parte inferior da área de trabalho de engenharia vai mostrar que uma sessão de depuração está ativa com os componentes do servidor ou os componentes do cliente do projeto em execução.

4. Quando o depurador dot NET está Anexado o sistema irá parar nos pontos de interrupção definidos e também irá parar automaticamente quando ocorrer qualquer NET Exception.

Veja no Editor de Código os Botões para Anexar e Desanexar depurador, executar, executar passo a passo criar pontos de interrupção mostrados no retângulo vermelho. Na parte inferior informação de status da depuração.



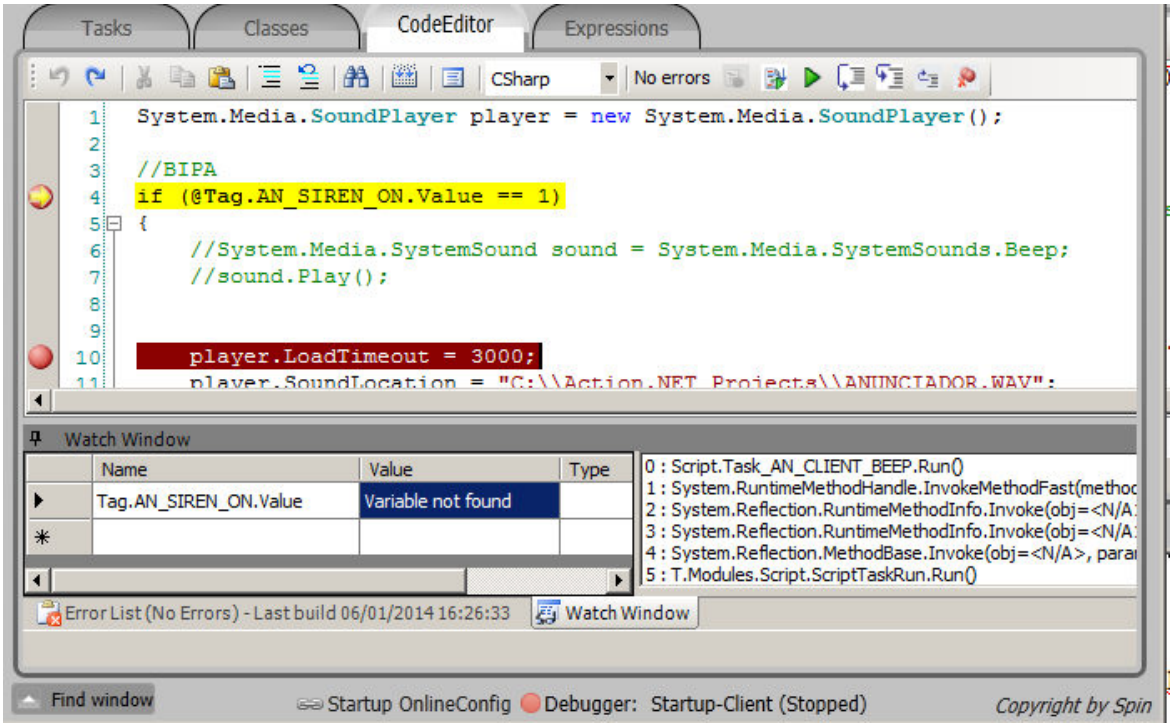
Pontos de interrupção, Passo a Passo e Watch

Para configurar um ponto de interrupção, abra o código desejado, selecione a linha e pressione o Insert Breakpoint na barra de ferramenta ou clique na barra vertical esquerda da janela de código.

Quando o sistema para em um ponto de interrupção, você pode executar passo a passo pressionando os botões de execução passo a passo ou mandar seguir pressionando o botão continuar.

A fim de inspecionar variáveis locais do dot NET tags ou objetos do projeto, você pode selecionar o texto no editor de script e, quando a execução for interrompida em um ponto de interrupção, o kit de ferramentas mostrará o valor atual da variável.

Você também pode adicionar. variáveis locais do dot NET ou objetos do projeto na janela Watch do próprio Code Editor (parte inferior) Ao adicionar tags ou objetos do projeto, você precisa usar os símbolos @, example@tag.tag1, para que o sistema distingua entre objetos de projeto, e variáveis locais do dot NET. Tenha em mente o Watch atualizado apenas quando a execução é interrompida. Se você quer ter os valores em tempo real para tags e objetos que você pode abrir a ferramenta de diagnóstico PropertWatch descrita em "Ferramentas para diagnose" na pagina 112.



Objeto Runtime Script

O namespace Script é o ponto de entrada para todos os objetos relacionados ao módulo de Scripts.

O objeto Script.Task lista todas as Tasks configuradas e suas propriedades em tempo de execução.

Os objeto **`Script.Class lista`** todas as Classes e suas propriedades em tempo de execução configurado.

Veja em http://www.spinengenharia.com.br/help/an-2014/runtime/index.html . para a referência de programação completa em objetos de tempo de execução.

Telas e Símbolos

As seções a seguir descrevem como criar a interface de usuário da aplicação:

- "Seleção e Criação de Telas" na pagina 147
- "Criação de Layouts" na pagina 152
- "O ambiente Desenhar (Draw)" na pagina 159
- "Ferramentas de desenho" na pagina 160
- "Dinâmicas e Animações" na pagina 166
- "Controles da interface do usuário" na pagina 176
- "Gerenciando Imagens e Símbolos" na pagina 184
- "Objetos runtime Display" na pagina 192

Seleção e Criação de Telas

No Action.NET a janela principal da tela do aplicativo é definida por um objeto denominado **Layout.** Um **Layout** define o tamanho da janela do aplicativo e o arranjo básico de componentes de visualização (Telas e partes de Telas) na aplicação em tempo real. Veja "Criação de Layouts" na pagina 152

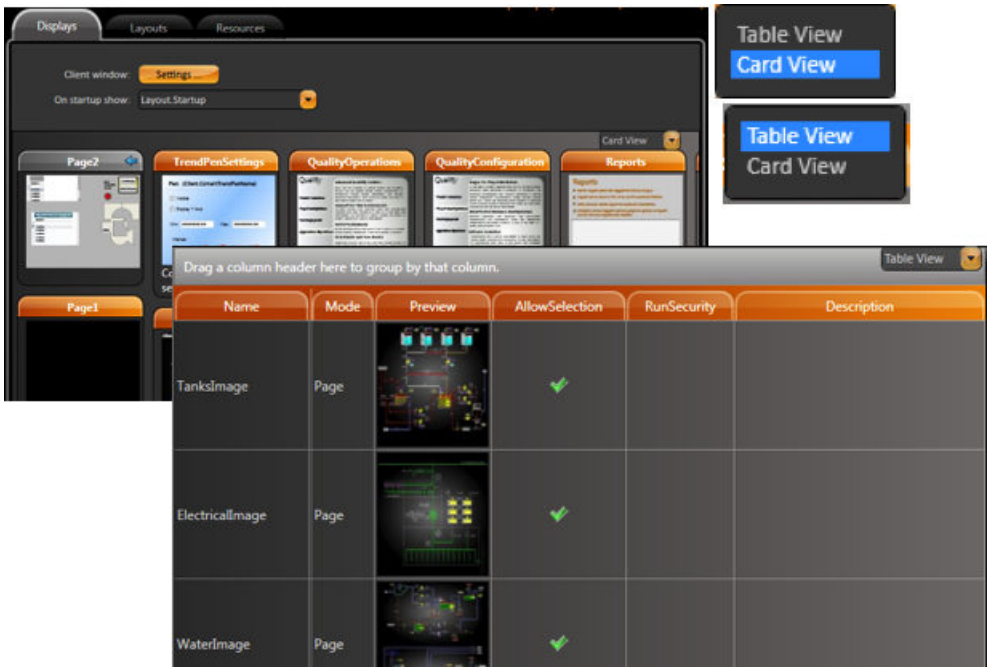
Telas (Displays) são componentes da aplicação que servem como interface entre o usuário e o processo monitorado. Cada Tela pode ser formada por várias partes ou telas estáticas ou dinâmicas. Uma tela pode conter vários elementos de visualização e comando, incluindo controles, áreas de Tela de dados, áreas estáticas e outros objetos. Estes componentes ou telas podem ser estáticos, ou seja, eles sempre são exibidos, como uma barra de menu, barra de ferramentas ou barra de status. Os componentes também podem ser trocados dinamicamente, dependendo do que o usuário clica ou seleciona.

Para selecionar Telas:

1. Vá para **Edit> Displays> Displays**.
2. Selecione a opção de visualização **Cardview** ou **TableView** na parte superior direita do DataGrid com a lista de Telas.
3. Veja a lista de Telas disponíveis aparecer no DataGrid.

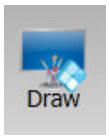
Coluna	Descrição
Name	Somente leitura. Configurado ao definir a Tela no ambiente **Draw.**
Mode	Somente leitura. Configurado ao definir a Tela no ambiente **Draw.**
Preview	Somente leitura. Mostra uma miniatura da tela.
AllowSelection	Selecione para permitir que os usuários possam selecionar a Tela através do seletor de páginas (PageSelector).
EditSecurity	Escolha quais tipos de usuários têm acesso para editar esta Tela.

Coluna	Descrição
RunSecurity	Escolha quais tipos de usuários têm acesso a esta Tela em tempo de execução.
Description	Digite uma descrição para esta Tela.



Operações em Telas

Para editar uma Tela, selecione-a no DataGrid na aba **Display** e pressione o botão DRAW.



Quando você está no ambiente DRAW, você também pode editar telas selecionando-as na combo-box na barra de ferramentas. que aparece para a execução de operações com as Telas.

- A caixa (combo-box) a esquerda mostra o nome da Tela atualmente carregada na área de trabalho. Abrindo esta caixa (clique na seta) aparece a lista com todas as telas existentes no projeto. Escolhendo-se uma na lista , esta será carregada na área de Desenho e passará ser a tela corrente para edição.
- O ícone em formato de disquete serve para solicitar salvamento das alterações;
- O ícone com o disquete e a figura abrindo a ficha serve para solicitar salvar a tela com outro nome (salvar como).

- O ícone com o sinal + deve ser utilizado para criar uma nova tela, vazia, e fechar, salvando-a a tela até então carregada. Após as alterações na nova tela aparece janela solicitando o nome a ser dado para a nova tela.
- O símbolo com o Monitor de vídeo permite mostrar uma visão prévia da tela, isto é mostrar como ficará a tela em tempo real.
- O ultimo ícone serve para fechar a tela corrente sem salvar as ultimas alterações

---

Criando uma nova tela

Para criar uma nova Tela

1. Vá para **Draw> Drawing** .
2. Clique em cuja ação é Fechar a Tela atual e Criar uma nova.
   - Se você não encontrar este botão na barra de ferramentas, faça a sua janela mais larga.
   - Será exibida a janela para criar uma nova tela.

3. Digite ou selecione as informações, conforme necessário.

Coluna	Descrição
Display Name	Digite um nome para a tela.

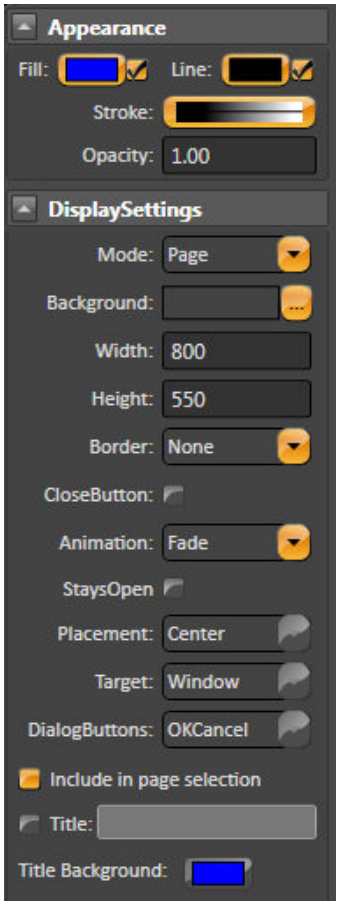
Coluna	Descrição
Description	Digite uma descrição para a tela.

iPad/iPhone iOS Target	**Nota**: Se você planeja usar essa Tela para usuários do iPad e do iPhone, não se esqueça de selecionar essa opção ao criar o display. Você não pode mudá-la mais tarde.
	• Selecione para ser possível usar essa Tela com iPads e iPhones. Ao selecionar esta opção, as configurações de Tela em outras partes da área Draw mostram apenas as opções que se aplicam ao Windows e iPads / iPhones.
	• Selecione o modo de Tela padrão para usuários de iPad e iPhone:
	  • Paisagem
	  • Retrato

4. Clique **OK**, para fechar a janela de criação da tela.

AVISO: Se a tela é para uso com um iPad ou iPhone, você deve selecionar a opção de iPad / iPhone iOS Target quando você criar o display. Você não pode alterar esta definição mais tarde.

5. No lado esquerdo da janela principal aparece uma área para o detalhamento dos atributos da tela, com duas partes: **Appearance** e **DisplaySettings**.



Os parâmetros de **Aparência** definem o estilo do "pincel" usado ao desenhar objetos na tela, incluindo as cores. Clicando na opção **Fill** (preencher) você poderá definir Cores, gradientes ou objetos para o preenchimento de uma figura gráfica.

Os parâmetros em **DisplaySettings** definem, os atributos da tela . As telas podem ser configuradas como PopUps, Diálogos, ou Telas normais.

Como qualquer objeto gráfico, as Telas podem ter cores e padrões diferentes, bordas, títulos e muito mais.

Uma característica especial é que você pode especificar uma transição de animação para usar quando trocar de uma tela para outra de forma semelhante ao que se usa nas apresentações do PowerPoint.

Se você tiver selecionado a opção de iPad / iPhone iOS alvo quando criou a exibição, nem todos os **DisplaySettings** estarão disponíveis.

6. Digite ou selecione as informações, conforme necessário, para a definição dos **DisplaySettings**.

Coluna	Descrição
Mode	Selecione o tipo de Tela que você deseja criar: • **Page** - É o modo default. Ao abrir uma tela do modo **Page** causa o fechamento automático da última página no layout atual e exibe na mesma área a ultima da lista da página em **Displays> Layouts**. Apenas telas em modo Page são exibidas no layout. • **Popup** - Uma pop-up é aberta na frente de todas as outros telas. Quando você abre uma nova página, por padrão, todas as pop-ups exibidas são fechadas. Controles em outras telas ficam disponíveis durante o aparecimento de pop-ups. • **Dialog** - Uma Tela diálogo é aberta como uma caixa de diálogo modal, que desativa os controles em todas as outras telas abertas até fechar a caixa de diálogo. Clicando OK executa o método OnOK no CodeBehind da Tela (display).
Background	Selecione a cor de fundo da tela.
Width	Introduza a largura da tela, em unidades do WPF (numero de pixels independentes de monitor). Certifique-se de levar em conta o tamanho do layout.
Height	Introduza a largura da tela, em unidades do WPF (numero de pixels independentes de monitor). Certifique-se de levar em conta o tamanho do layout.
Border	Selecione o tipo de borda da tela.
CloseButton	Selecione se deve exibir o botão Fechar da tela. Disponível apenas para Popups e diálogos.
Animation	Selecione o tipo de animação (Nenhum, fade,slide, scroll) a ser usada ao mostrar esta tela..
Show on PageSelector Object	Marque a opção para incluir esta Tela no Seletor de Telas que permite aos usuários acessar diretamente uma Tela na aplicação, sem necessidade de incluir botões de navegação nas telas.
iPad/iPhone iOS target	Somente leitura. Mostra se você selecionou a opção tela para ser usada em iPad / iPhone iOS quando criou a Tela.
Placement	Selecione onde, dentro do Layout, que você deseja colocar esta tela. Disponível apenas para Popups e diálogos.
Target	Selecione se o "Placement" definido é em relação à posição do mouse ou em relação à janela inteira. Disponível apenas para Popups e diálogos.
DialogButtons	Selecione os botões de controle que você quer na caixa de diálogo. Disponível apenas para diálogos.
Title	Digite um título que vai aparecer na parte superior da tela.
Title Background	Selecione uma cor para o fundo do título.
Stays Open on Page Change	Deixe a janela pop-up aberta quando o usuário clica em algo que abre uma tela diferente. Disponível apenas

Coluna	Descrição
Stays Open After Losing Focus	Deixe a janela pop-up aberta quando o usuário clica em uma outra. Tela. Disponível apenas para Popups.
Left	Digite, a distancia entre a margem esquerda do Layout e a posição em que o lado esquerdo da tela deve aparecer, em unidades WPF (pixels independente de monitor). Disponível somente para Layouts Canvas.
Top	Digite, a distancia entre o topo do Layout e a posição em que o topo da tela deve aparecer, em unidades WPF (pixels independente de monitor). Disponível somente para Layouts Canvas.

7. Use as ferramentas de desenho, nas barras vertical aà esquerda e horizontal inferior para criar, e posicionar objetos de visualização na Tela
   - Selecione um dos botões na barra de ferramentas vertical. Coloque o cursor na área de exibição, em seguida, clique e segure o botão esquerdo do mouse enquanto arrasta o cursor em na área da Tela recém crida.
   - Para ver um menu de opções, clique com o botão direito do mouse sobre o objeto desenhado. O menu de contexto fornece ações que são específicas para o objeto selecionado.
   - Para selecionar vários objetos, pressione **Shift** + clique com o botão esquerdo do mouse em cada objeto.

   AVISO - Ao selecionar vários objetos, o último objeto selecionado é o "mestre": as propriedades dele são as mostradas no painel à esquerda. Se você, em seguida, alinhar os objetos, o alinhamento é baseado no mestre. Se você alterar outras propriedades, as propriedades são alteradas para todos os objetos selecionados.

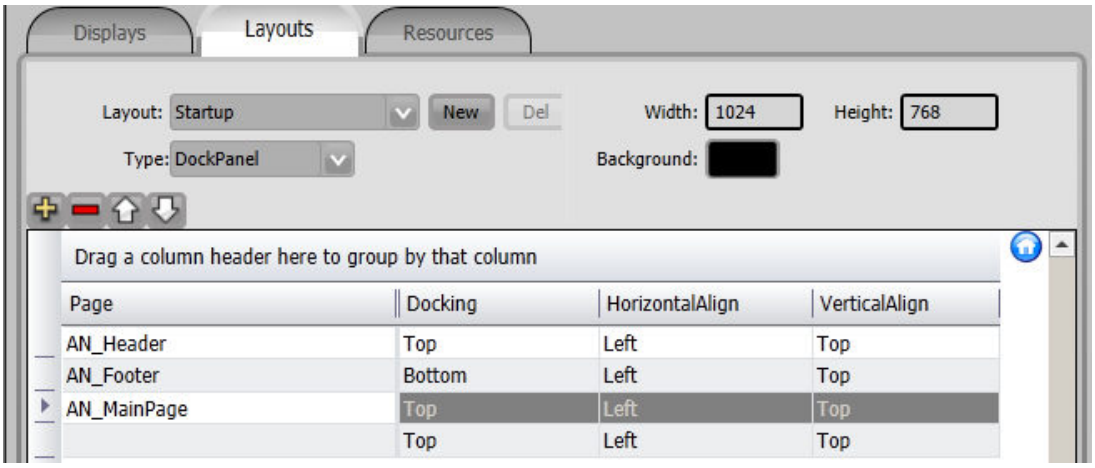
   - A barra de ferramentas horizontal (na parte inferior da aba **Drawing**) contém botões para agrupar, combinar, alinhar e bloquear os objetos selecionados.
   - Para mais informações sobre as ferramentas de desenho, consulte "Ferramentas de desenho" na pagina 160.
8. Clique em **Save Display**.
9. Continue na seção "Criação de Layouts" na pagina 152.

Criação de Layouts

Um layout define o tamanho da janela do aplicativo e o arranjo básico de componentes de visualização na aplicação de tempo real. Certifique-se de planejar como você deseja utilizar as Telas. Por exemplo, considere se você quer uma barra de menu, barra de ferramentas ou outros elementos para exibir sempre no topo da janela da aplicação.

Muitas vezes você pode precisar de um único layout para enquadrar qualquer das telas com toda a informação na aplicação. Usa-se aba **Layouts** no ambiente de **Displays** para a definição de layouts.

NOTA - A última Tela listada na aba Layout é a única que é substituída quando você troca de Tela. Por exemplo, os novos projetos incluem um layout padrão chamado Startup. Veja na figura que há outras duas telas, ou componentes, neste Layout: AN_Header (uma barra de ferramentas) e AN_Footer (um rodapé). A tela AN_MainPage que está listada na última linha do DataGrid é a área que, por default, será trocada quando você solicita para a aplicação Abrir outra tela.

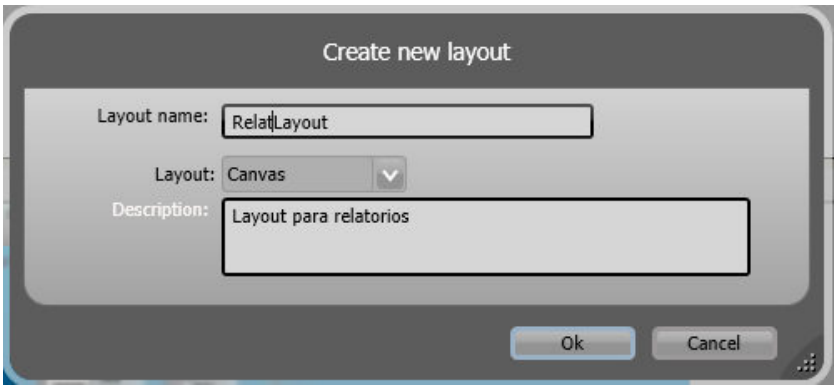


Na figura, os botões acima do DataGrid à esquerda na barra servem, respectivamente, para:

- Adicionar nova linha e descrever posição da tela
- Retirar a linha selecionada
- Subir a linha selecionada
- Descer a linha selecionada

Para criar um novo layout:

1. Vá para **Edit> Displays> Layouts** .
2. Clique o botão **New**. Será mostrada a janela para a criação de **Layouts.**



3. Digite ou selecione as informações, conforme necessário.

Coluna	Descrição
Layout name	Entre com um Nome para o layout
Layout	Selecione o tipo de Layout sendo criado: • **DockPanel** - Selecione este tipo para posicionar telas em relação ao layout e a outras telas. • **Canvas -** Selecione para posicionar telas em uma posição absoluta, introduzindo a posição em unidades WPF (pixels independentes do monitor).
Description	Digite uma descrição desse layout.

4. Clique em **OK**. A janela de criação de Layouts será fechada
5. No canto superior direito da aba Layouts, digite ou selecione as informações, conforme necessário:

Coluna	Descrição
Width	Digite a altura, em unidades WPF (pixels independentes do monitor), do layout. Esta é a altura da tela da aplicação em tempo de execução.
Height	Digite a largura, em unidades WPF (pixels independentes do monitor), do layout. Esta é a largura do pedido de execução
Background	Selecione uma cor para fundo deste layout.

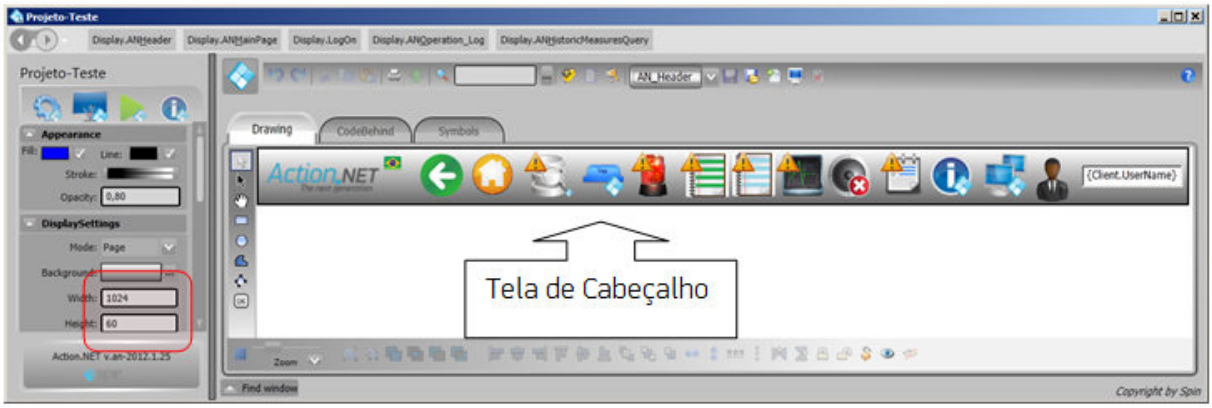
6. 6. Clique em **Adicionar** linha para adicionar uma nova linha para posicionar uma Tela.
   - As Telas que você adiciona são as primeiras telas que o layout utilizará.
7. Digite ou selecione as informações, conforme necessário.

O exemplo de Layout StartUp

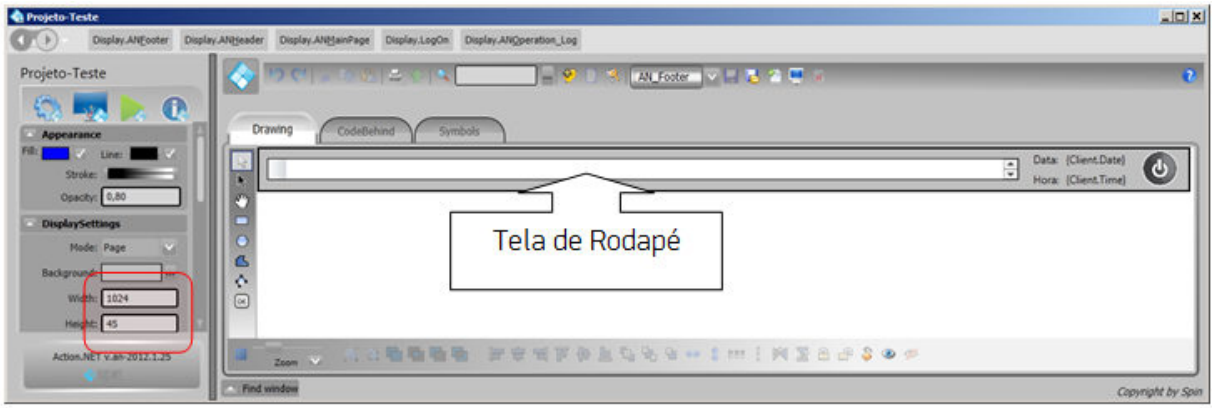
O **Layout Startup,** mostrado no DataGrid acima, vem no projeto Default do Action.NET. A seguir informações sobre cada uma destas áreas:

Trata-se de um esquema definindo uma tela de cabeçalho outra de rodapé e a tela principal na parte central do monitor, para que sejam mostradas como uma única tela pode ser definido por um Layout conforme apresentado abaixo no layout **Startup**:

- AN_Header: é o cabeçalho da tela ajustado no topo à esquerda. Como a dimensão do layout é 1024 x 768, para o cabeçalho ocupar toda a parte horizontal superior do layout deverá ter comprimento (width) de 1024. No caso desta aplicação, sua dimensão é 1024 x 60.



- AN_Footer: é o rodapé da tela, ajustado na parte inferior da tela à esquerda. De forma similar ao cabeçalho, deverá ter comprimento de 1024 para ocupar toda a parte inferior da tela. No caso desta aplicação sua dimensão é 1024 x 45.



- AN_MainPage: será o miolo do layout com as diversas telas de processo que serão utilizadas. Nesta aplicação, considerando que o cabeçalho tem altura (Height) de 60 pontos e o rodapé 45, as telas de miolo deverão ter dimensão vertical de 768 – 60 – 45 = 663. Assim, todas as telas inseridas no miolo deste layout deverão ter a dimensão de 1024 x 663.

NOTA - Quando em um projeto existe mais de um layout, antes de se abrir uma tela deve-se abrir seu layout e a tela sempre será colocada no último nível do layout (no caso acima nível 2). A título de exemplo, abaixo é mostrado um script abrindo um layout e uma tela:

```
@Layout.AN_FilterScreens.Open();
@Client.OpenDisplayAtIndex("AN_AlarmsSummary",@Script.Class.AN_LastPage.GetInd
exFILTERSCREENS_MAIN());
```

Coluna	Descrição
Page	Selecione uma Tela que você deseja incluir no layout. Somente Telas tipo Page podem ser utilizadas para posicionamento em Layouts.
Docking	Selecione a margem da área principal de visualização da aplicação, para ser o local de fixação (docagem) para esta Tela: • Left (esquerda), Top (topo), Right (direita), Bottom (inferior)
HorizontalAlign	Selecione o alinhamento horizontal desta tela no Layout: • Left (esquerda), Center (centro), Right (direita)
VerticalAlign	Selecione o alinhamento vertical desta tela no Layout: • Top (topo), Center (centro), Bottom (inferior)

Configurações da área de Visualização

As configurações da área de visualização controlam como o aplicativo deve exibir as telas nos monitores dos computadores clientes, isto é que utilizam os módulos de visualização do Action.NET.

Para esta configuração há botões e controles na parte superior do ambiente Display, mostrados na figura abaixo.

Estas configurações são diferentes para clientes normais MS-Windows daquelas necessárias para clientes utilizando o iOS dos tablets e iPhones da Apple. As próximas seções detalham estas configurações.

Clientes MS-Windows ("Desktop")

Para definir as configurações do cliente de desktop:

1. Vá para **Edit>Displays>Displays** .
2. Clique em **Clients** para configurar desktops que usam Windows. Aparece a janela para configuração

3. Digite ou selecione as informações, conforme necessário e clique em **OK**

Coluna	Descrição
Maximize When Open	Selecione para maximizar a janela do cliente de forma a se ajustar ao monitor.
Stretch	Escolha como o layout deve se estender no monitor cliente quando usando o aplicativo: • None - Layout não deve se redimensionar: Deve ser mostrado o tamanho exato que foi configurado para o layout. Neste caso recomenda-se selecionar a opção de incluir a barra de rolagem. • Fill (Preencher) - O layout será redimensionado para preencher completamente o espaço disponível no monitor do cliente., sem levar em conta a proporção do layout original. • Uniforme - O layout será redimensionado proporcionalmente ao tamanho do monitor do cliente, mantendo a proporção do layout. original. • UniformToFill - O layout redimensionado tanto para manter a proporção como também para preencher completamente o espaço disponível no monitor do cliente. Isto pode resultar em cortar algumas partes da disposição.
Scrollbar	Selecione para visualização da barra de rolagem na janela do cliente.

Coluna	Descrição
Title	Digite um título para a janela do cliente.
MinimizeBox	Selecione para exibir o botão padrão de minimizar.
MaximizeBox	Selecione para exibir o botão padrão de maximização.
CloseBox	Selecione para exibir o botão padrão de fechar
ResizeBox	Selecione para exibir o puxador padrão de redimensionamento (canto inferior direito)
Menus	Selecione quais menus devem ser mostrados • File • Tools • Security
Mouse Cursor Visible	Selecione para o ponteiro do mouse ser visível.
OnScreen KeyPad	Selecione para exibir um teclado numérico na tela.
Disable Web Comments	Selecione para proibir usuários do aplicativo de enviar comandos para o PLC
When Mouse Is Over Command Areas	Selecione o que mostrar: • Mostrar bordas do objeto • Alterar o Cursor - Selecione o tipo de cursor.
Verify Tag Quality on Client Displays	• On undefined quality show - (Quando a qualidade for indefinida) Selecionar a opção e o caractere que deve ser mostrado quando a qualidade dos dados vindos de campo for indefinida. • On OPC Bad quality show - Selecionar a opção e o caractere que deve ser mostrado quando o servidor OPC indica que a qualidade dos dados é ruim.
Share Logged User at Same Computer	Os usuários podem executar múltiplas instâncias do aplicativo no mesmo computador. O usuário pode precisar logar como um usuário diferente de vez em quando. Selecione essa opção para alterar automaticamente o usuário conectado em todas as instâncias em execução do aplicativo no mesmo computador.

4. Digite o número de monitores que o visualizador Cliente vai usar.
5. Selecione o monitor e o Layout a ser mostrado quando o aplicativo é iniciado.

Cliente iOS

Se necessário, faça também as configurações da visualização para os Cliente iOS.

Para configura a área de trabalho para Clientes iOS:

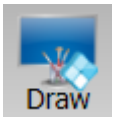
1. Pressione o botão **iOS Device**. Aparece a janela para configuração para iOS client.

2. Digite ou selecione as informações, conforme necessário e clique em OK .

Coluna	Descrição
Disable Commands	Selecione para proibir usuários do aplicativo de enviar comandos para o campo
iPad Initial Page	Selecione a Tela que os usuários do iPad devem ver quando iniciar a aplicação no iPad.
iPhone Initial Page	Selecione a Tela que os usuários do iPhone devem ver quando iniciar a aplicação no iPhone.
Stretch Fill	Selecione para redimensionar o aplicativo de modo a preencher a tela do dispositivo.

O ambiente Desenhar (Draw)

O ambiente de trabalho **Desenhar (Draw)** disponibiliza todas as ferramentas para a construção de telas gráficas para a aplicação.



Para entrar neste ambiente clique o ícone de **Desenhar**.

No Action.NET o ambiente **Desenhar** possui três abas:

Design (Desenho) onde são mostradas e podem ser editadas as telas. Na aba de Desenho, existem uma barra de ferramentas vertical no lado esquerdo da área de trabalho e uma barra horizontal na parte de baixo da mesma área, orientadas a tarefas sobre os objetos dentro de uma tela.

CodeBehind para a escrita de Scripts relacionados à tela. Nesta aba há um Editor de Código fonte e já aparecem os textos dos protótipos dos procedimentos que são ativados por eventos básicos sobre telas. Se necessário estes textos devem ser preenchidos com códigos para modificar o procedimento padrão da tela.

Symbols com a biblioteca de símbolos gráficos disponíveis no projeto. Neste grid pode-se editar atributos de símbolos. Para criar um símbolo, usar o botão direito do mouse sobre um objeto em uma tela.

Code Behind - Scripts de Telas

Use a aba **CodeBehind** para definir um conjunto de funções relacionadas com a Tela. Você pode escrever código tanto em VB.Net como em CSharp. Você também pode alternar entre os dois. Se você alterar a seleção da linguagem, o sistema converterá automaticamente o código existente para o correspondente na linguagem selecionada.

Se você precisar de referências a outras partes de código também criados por você, use **Run > Build > References** para estabelecer estas referências.

As funções colocadas no CodeBehind podem ser executadas ao abrir ou fechar uma tela, ou quando a tela já está aberta, dependendo de como você configura o código. Você pode usar CodeBehind para definir métodos de tratamento dos cliques do mouse sobre a tela ou de entrada de comandos para serem executados em telas específicas.

Para Telas do tipo diálogo, use o método pré definido `DialogOnOK`, que é chamado quando o botão **OK**, também pré definido na caixa de diálogo, é pressionado. Se este método retornar `True`, o diálogo é fechado, se retornar `False`, o diálogo continua aberto. Este método é comumente usado para garantir a validação de dados na caixa de diálogo (que deve então solicitar ao usuário para corrigir entradas incorretas antes de fechar o diálogo).

O código por trás das telas tem os seguintes métodos pré-definidos:

- DisplayOpening () - Executado quando a tela está abrindo.
- DisplayIsOpen () - Chamado em intervalos regulares de tempo enquanto a tela está aberta.
- DisplayClosing () - Executado quando a tela está se fechando.
- DialogOnOK () - Chamado quando o botão OK de uma tela de diálogo é pressionado: - Retornando 1 permite o diálogo ser Fechado pelo sistema. Retornando 0 impede o diálogo de ser fechado.

Você pode adicionar suas próprias variáveis e métodos dot NET neste texto de código.

NOTA - Como as Telas nos Clientes são projetadas para funcionar em ambientes distribuídos e web, recomendamos evitar o uso de funções que não permitam a execução em "confiança parcial", ou que se referem a caminhos de arquivos físicos.

Ferramentas de desenho

A aba **Drawing** tem os seguintes barras de ferramentas e controles para criar, formatar e configurar as Telas:

- **Barra Vertical** - No lado esquerdo da aba Drawing. Use estes botões para desenhar formas, adicionar botões, e criar janelas especiais. Para detalhes sobre estes botões, ver Tabela 1 .
- **Barra Horizontal** - Na parte inferior da aba Drawing. Use estes botões para agrupar, combinar, alinhar e fixar os objetos selecionados. Para mais informações, ver Tabela 2.
- **Aparência, Dynamics, DisplaySettings** e outras configurações - No painel a esquerda da aba, abaixo dos botões do menu principal. Estas configurações variam de acordo com as características e funcionalidades do tipo de objeto selecionado.

Drawing - Barra de ferramentas vertical

A tabela 1 a seguir apresenta os ícones da barra de ferramentas Vertical do espaço Drawing e as suas funcionalidades.

Tabela 1: Barra de Ferramentas Vertical

Botão	Descrição
Selection Tool	• Clique em um objeto para selecioná-lo. • CTRL + clique para selecionar vários objetos (e grupos de objetos), um de cada vez. Mantenha pressionada a tecla CTRL enquanto clica em cada objeto. • Shift + clique para alternar objeto selecionado entre mais de um objeto selecionado. • Clique em uma área aberta da tela, em seguida, selecione um grupo de elementos, destacando os elementos enquanto arrastando o mouse e mantendo pressionado o botão esquerdo do mouse. • Clique duas vezes em um objeto para abrir a janela de configuração Dynamics, que permite definições para propriedades de objetos dinâmicos.

Direct Selection Tool	• Use esta ferramenta para selecionar um objeto dentro de um grupo (e modificar suas propriedades). Clique no objeto para selecioná-lo.
	• Você também pode adicionar, remover e modificar os pontos em um polígono com a Direct Selection Tool:
	  ◦ Para mover o ponto, selecione-o clicando no ponto e segurando o botão esquerdo do mouse pressionado. Arraste o ponto para sua nova posição.
	  ◦ Clique duas vezes em um ponto para adicionar um novo ponto adjacente ao ponto selecionado.
	  ◦ Para excluir um ponto selecionado pressione o botão direito.

Hand Tool	Use a ferramenta Mão para modificar a janela de visualização clicando no fundo da Tela e mantendo pressionado o botão esquerdo do mouse em seguida, mudar a tela para a posição desejada.
Geometric objects tools	Use o Botão direito do mouse para encerrar o uso de cada ferramenta.. Para adicionar, alterar e remover os pontos depois de criar um polígono ou polilinha, use a ferramenta de seleção direta.

Tabela 1: Barra de Ferramentas Vertical

Botão	Descrição
Rectangle	Cria um objeto retângulo.
Ellipse	Cria um objeto elipse.
Polygon	Cria um objeto polígono.
Polyline	Cria um objeto polilinha.
Button	Cria um objeto botão.
Text Output	Cria um objeto de saída de texto.
Text Box	Cria um objeto de entrada/saída de texto. Para vincular o objeto de texto de I / O com uma tag, clique duas vezes no objeto de texto de I / O. Na janela de configuração dinâmica selecionar TextIo dinâmico.
CheckBox	Cria um objeto do tipo "check box". Clicando com o botão direito do mouse tem-se acesso a criação dos seguintes outros objetos: Cria um objeto radio button Cria um objeto combo box. Cria um objeto list box. Cria um objeto senha. Cria um objeto "Pega data" Cria um objeto date/time text box. Cria um objeto rótulo. Cria um objeto controle deslizante. Depois de selecionar o objeto deste menu, ele se torna o objeto default para este botão na barra vertical.
Open Symbol Library	Clicando este botão mostra a Biblioteca de Símbolos que inclui tanto os símbolos pre definidos como os definidos pelo usuário.

Tabela 1: Barra de Ferramentas Vertical

Botão	Descrição
Web Browser	Cria um objeto Web Browser. Clicando com o botão direito do mouse tem-se acesso a criação dos seguintes outros objetos: Cria um objeto Page Selector, onde os usuários podem ir para escolher diretamente para uma outra Tela na aplicação. Cria um objeto Report Viewer . Cria um objeto n XPS Viewer. Cria um objeto Circular Panel. para mais informação, veja em "Controles da interface do usuário" na pagina 176. Cria um objeto Gráfico de Pizza. Para mais informação veja em "Configurando um Gráfico de Pizza" na pagina 177. Cria um objeto Gráfico de Barra. Para mais informação, veja em "Configurando um Gráfico de Barras (BarChart)" na pagina 177. Cria um objeto Calculadora Cria um objeto janela filha (Child Window). Cria um componente WPF externo. Depois de selecionar o objeto deste menu, ele se torna o objeto default para este botão na barra vertical.
Alarm Window	Cria um objeto Janela de Alarme. Posicione a janela de alarme na tela e faça um duplo clique sobre ela para configurar seus parâmetros. Para mais informação veja em "Configurando uma Janela de Alarmes" na pagina 178.
Trend Window	Cria um objeto Janela de Tendência. Posicione a janela de tendência na tela e faça um duplo clique sobre ela para configurar seus parâmetros. Para mais informação veja em "Configurando a Janela de Tendência" na pagina 179.
DataGrid Window	Cria um objeto janela DataGrid. Posicione a janela DataGrid na tela faça um duplo clique sobre ela para configurar seus parâmetros. Para mais informação veja em "Configurando uma Janela DataGrid" na pagina 181.

Drawing - Barra de ferramentas Horizontal

A tabela 2 a seguir apresenta os ícones da barra de ferramentas Horizontal do espaço Drawing e as suas funcionalidades.

Tabela 2: Barra de Ferramentas Horizontal

Botão	Descrição
	Definição de Grade
	Zoom da tela
	Agrupar
	Desagrupar
	União de desenhos
	Intercecção de desenhos
	Exclusão - Exclui partes não comuns
	Ou exclusivo - Fica somente com partes não comuns
	Alinhar a esquerda
	Alinhar horizontalmente no centro
	Alinhar a direita
	Alinhar topos
	Alinhar verticalmente no centro
	Alinhar bases
	Mover para a frente
	Mover para trás
	Modificar largura
	Modificar altura
	Espaçar igualmente na horizontal
	Espaçar igualmente na vertical
	Girar Horizontalmente

Tabela 2: Barra de Ferramentas Horizontal (Continued)

Botão	Descrição
	Girar Verticalmente
	Impedir mudança de posição
	Liberar movimentação
	Liberar movimentação de todos objetos impedidos.
	Mostrar todos os objetos
	Esconder objetos selecionados

Propriedades dos Controles

Na área **DRAW**, no lado esquerdo é mostrado o painel de configuração de propriedades de telas e de controles de visualização. As seções seguintes mostram detalhes sobre estas configurações.

Veja em "UId e Texto" na pagina 165, a configuração do texto e Identificador.

Veja em"Dinâmicas e Animações" na pagina 166, configuração de Dinâmicas.

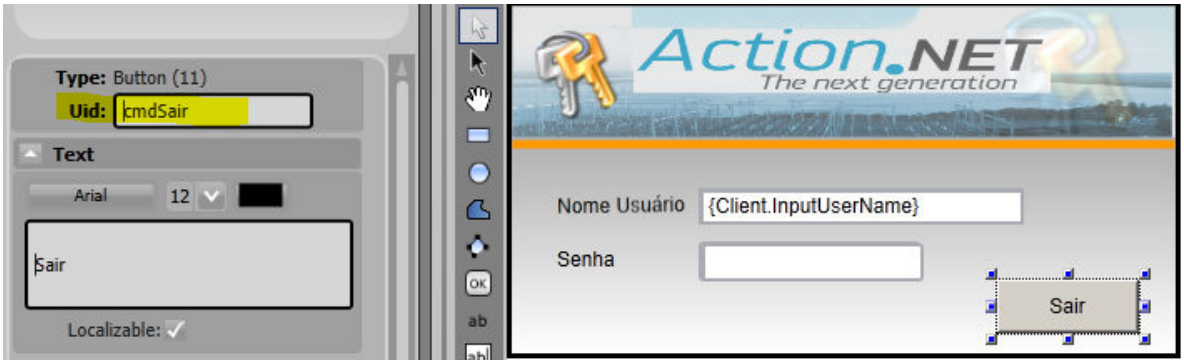
Veja em "Aparência e Transformação" na pagina 182, sobre Aparência.

Veja em "Alteração de vários controles" na pagina 183, para operações de troca de tags.

UId e Texto

Quando se cria um objeto ou um símbolo novo na tela, ele recebe um número que o identifica univocamente: O **<Uid>**. Este `Uid` pode ser rebatizado por um String qualquer, o que tem muita utilidade quando se necessita, por exemplo, escrever código de Script (CodeBehind) envolvendo o controle.

Como exemplo, apresenta-se um Botão usado na janela de Login. Este objeto teve o seu numero alterado para `cmdSair`, para facilitar seu entendimento dentro do código, conforme se pode ver na figura a seguir:



Text é a área onde se pode especificar uma legenda ou titulo para um controle que permita esta funcionalidade. Pode se especificar a Fonte, o tamanho e a cor do texto, além do texto propriamente.

A opção **Localizable** deve ser marcada em caso de se estar utilizando dicionários de linguagens no projeto, e se desejar que este texto seja alterado de acordo com o dicionário.

Dinâmicas e Animações

A área de configuração **Dynamics** permite que você configure as mudanças em tempo real na aparência de um objeto, sua posição, seu tamanho, suas cores, o valor mostrado pleo objeto, a ação que o objeto deve apresentar quando um usuário o clica, e outras coisas. Esse comportamento dinâmico é configurado através da criação de elos entre as propriedades do objeto e as propriedades das Tags ou outras propriedades em tempo de execução do projeto.

Em alguns sistemas, a denominação animação é usada para se referir a essas mudanças dinâmicas em tempo de execução. No Action.NET usa-se o termo Dynamics para distingui-las claramente dos recursos de animação fornecidos pelo WPF. As animações WPF também se referem a alterações nas propriedades gráficas dos objetos executando nos monitores, mas usando temporizadores e outros status de objetos para dirigir a animação, e não os valores da base de dados em tempo real.

Quando você clicar duas vezes em um objeto, ou pressionar o botão Dynamics, ou selecioná-lo no menu de contexto do botão direito, é apresentada uma lista das dinâmicas que podem ser aplicadas no objeto selecionado.

A Tabela 3 lista as dinâmicas disponíveis.

Tabela 3: Dinâmicas e Animações

Configuração	Descrição
Action	Executa ações e comandos acionados pela interface do usuário.
Shine	Altera a aparência objeto dinamicamente.
TextIO	Dinâmica de Entrada e saída de textos
HyperLink	Abre um hiperlink.
Security	Define as permissões do objeto para executar ações, em tempo de execução
FillColor	Muda dinamicamente a cor de preenchimento do objeto

Tabela 3: Dinâmicas e Animações

Configuração	Descrição
LineColor	Muda dinamicamente a cor da linha do objeto
TextColor	Muda dinamicamente a cor de texto do objeto
Bargraph	Dinâmica de barra gráfica
Visibility	Altera dinamicamente a visibilidade e opacidade objeto .
MoveDrag	Move dinamicamente o objeto.
Scale	Muda dinamicamente o tamanho do objeto.
Rotate	Gira o objeto dinamicamente.
Skew	Distorce o objeto dinamicamente.
TextOutput	Cria a dinâmica de saída de texto.

Action Dynamic

Tabela 4: Configuração da dinâmica Action

Configuração	Descrição
Action	Executa a ação desencadeada pela interface do usuário.

Tabela 4: Configuração da dinâmica Action

Configuração	Descrição
Event	Escolha um dos eventos do mouse. Mais do que um evento pode ser selecionado para cada ação. Por exemplo: Uma ação para o evento MouseLeftButtonDown e outra ação para MouseLeftButtonUp.

Action	Selecione uma ação para o evento determinado:
	• Nenhum - Nenhuma ação.
	• SetValue - Define o valor do objeto.
	  • Object - O objeto que vai receber o valor.
	  • Valor - O valor que será passado para o objeto.
	• ToggleValue - Alterna o valor do objeto. Se o valor atual do objecto é zero, o valor vai para 1. Se o valor atual do objecto é diferente de zero, o valor vai para 0.
	  • Object - O objeto que será alternado.
	• OpenDisplay - Abre um display.
	  • Display- O nome da Tela que será aberta.
	• CloseDisplay-Fecha um display.
	  • Display-O nome da Tela que será fechada.
	• OpenLayout - Abre um layout.
	  • Layout - O nome do layout que será aberto.
	• RunScript - Executa um script que deve ser colocado na aba Display> CodeBehind. Execute uma das seguintes ações:
	  • Digite o nome do novo método e clique Novo .
	  • Escolha um dos métodos existentes no comboBox.
	• RunExpressions - Executa a expressão dada.
	  • Expressão - Digite a expressão. Por exemplo: Tag.a + 1, ou Tag.a + Tag.b ou Math.cos (Tag.angle) * Math.PI.
	  • Resultado (opcional) - Digite o tag ou a propriedade que irá receber o valor da expressão.

Exemplos de Run Expresssions:

- Somar dois valores e passar o resultado para outra tag.

```
Expressão-Tag.quantity1 + Tag.quantity2.
Resultado-Tag.totalQuantity.
```

- Incrementar um tag.

```
Expressão-tagCounter + 1.
Resultado-tagCounter.
```

- Incrementar um tag (0 - 10).

```
Expressão- If (tagCounter <10, tagCounter + 1,0).
Resultado:-tagCounter.
```

Para mais informações, consulte "Configurando Expressões" na pagina 141.

Dinâmica de Brilho (Shine Dynamic)

Tabela 5: Configuração da dinamica de Brilho

Configuração	Descrição
Shine	Altera a aparência do objeto dinamicamente.
IsMouseOver	Digite um tag que receberá o OverValue ou o NotOverValue.
OverValue	O valor de IsMouseOver quando o mouse está sobre o objeto.
NotOverValue	O valor de IsMouseOver quando o mouse não está sobre o objeto.
Mouse Over Appearance	A aparência do objeto quando o mouse está sobre ele. • Opacidade - A opacidade do objeto (0 = transparente, 1 = opaco). • Scale - O tamanho do objeto (0.5 = metade, 1 = o mesmo tamanho, 1,5 = um ano e meio, 2 = dobro do tamanho). • OuterGlow - Define a cor do OuterGlow, deve-se marcar a seleção para ativar ou desativar este efeito. • TextColor - Define a cor do texto, deve-se marcar a seleção para ativar ou desativar este efeito.
Mouse Not Over Appearance	A aparência do objeto quando o mouse não está sobre ele: • Opacidade - O opacidade do objeto(0 = transparente, 1 = opaco). • Scale - O tamanho do objeto (0.5 = metade, 1 = o mesmo tamanho, 1,5 = um ano e meio, 2 = dobro do tamanho).
Is Selected Appearance	A aparência do objeto quando o mesmo está selecionado: • IsSelected - Define se o objeto está selecionado • Opacidade - O opacidade do objeto(0 = transparente, 1 = opaco). • Scale - O tamanho do objeto (0.5 = metade, 1 = o mesmo tamanho, 1,5 = um ano e meio, 2 = dobro do tamanho).
Scale Reference	Centro Esquerda Para cima Para direita Para baixo

Dinâmica Text I/O

Tabela 6: Configuração da dinâmica Text I/O

Configuração	Descrição
TextIO	Dinamica de entrada e saida de texto. Se o texto é um valor de Tag ou uma propriedade, ele deve estar entre colchetes. Por exemplo: {Tag.analogInt1}.
Binding Mode	Associar um objeto com uma tag: • TwoWay - Permite entrada e saída. • InputOnly - Somente permite entrada de tetxo (o valor do tag atual não é mostrado, mas novos valores podem ser inseridos). • OutputOnly - Somente saída de texto é permitida.
Object or Expression	Objeto (entrada) ou Expression (somente saida) que está conectada com a caixa de texto.
DesignModeCaption	O valor mostrado no modo de engenharia: • • ShowObjectNames - O conteúdo do campo de texto é mostrado exatamente como ele é. • • ShowPlaceHolders - São mostrados carcteres # # # , o número de caracteres é definida pelo campo MaxLength.
Input Range	Define o intervalo numérico para o valor inserido.
MaxLength	Define o número máximo de caracteres.

Dinamica de Hyperlink

Tabela 7: Configuração da dinâmica de Hyperlink

Configuração	Descrição
HyperLink	Abre um hyperlink.
HyperLinkType	Selecione o tipo de hyperlink: • http • ftp • file • mailto • telnet
Url	Configurar o URL para abrir om hiperlink

Dinâmica de Security (Segurança)

Tabela 8: Configuração da Dinâmica Security

Configuração	Descrição
Security	Defina as permissões do objeto em tempo de execução:
Disable	Digite um Tag, uma propriedade, ou uma expressão que retorna um valor. Considerando o valor resultante: • Zero - objeto será habilitado. • Maior que zero - o objeto será desativado. Para mais informações, consulte "Configurando Expressões" na pagina 141.
Verify Permissions	Quando selecionado, apenas os grupos de permissões escolhidos, podem acessar o objeto.
Confirm Message	Mostra um diálogo de confirmação antes de tomar alguma ação: • textBox - Digite a mensagem que irá aparecer na caixa de diálogo. • checkBox - Ativa ou desativa a Confirmar Mensagem.

Dinâmica Fill, Line e Text Color

Tabela 9: Dinâmica Fill, Line e Cor de Texto

Configuração	Descrição
FillColor	Muda a cor de preenchimento do objeto dinamicamente.
Expression	O valor usado para a dinâmica FillColor.
Change Color	• UsingLimits - A cor resultante é determinada quando o valor for igual ou superior a um dos limites. • AbsoluteValue - A cor será o valor da expressão. O valor deve ser um nome válido de cor ou numero hexadecimal para cores. Por exemplo: "White" ou "# FFFFFFFF" Exemplo Limites: 1 - Red 10 - Blue Quando o valor é 0, o objeto terá sua própria cor (a dinâmica de preenchimento cor não vai fazer nada) Quando o valor é de 1 a 9, o objeto terá a cor vermelha. Quando o valor é superior a 10, o objecto terá a cor azul.
LineColor	Muda a cor da linha objeto dinamicamente:.
Expression	O valor usado para a dinâmica LineColor.

Tabela 9: Dinâmica Fill, Line e Cor de Texto

Configuração	Descrição
Change Color	• UsingLimits - A cor resultante é determinada quando o valor for igual ou superior a um dos limites. • AbsoluteValue - A cor será o valor da expressão. O valor deve ser um nome válido de cor ou numero hexadecimal para cores. Por exemplo: "White" ou "# FFFFFFFF" Exemplo Limites: 1 - Red 10 - Blue Quando o valor é 0, o objeto terá sua própria cor (a dinamica de LineColor não vai fazer nada) Quando o valor é de 1 a 9, o objeto terá a cor vermelha. Quando o valor é superior a 10, o objecto terá a cor azul.
TextColor	Muda a cor do texto objeto dinamicamente:.
Expression	o valor usado para a dinâmica TextColor.
Change Color	• UsingLimits - A cor resultante é determinada quando o valor for igual ou superior a um dos limites. • AbsoluteValue - A cor será o valor da expressão. O valor deve ser um nome válido de cor ou numero hexadecimal para cores. Por exemplo: "White" ou "# FFFFFFFF" Exemplo Limites: 1 - Red 10 - Blue Quando o valor é 0, o objeto terá sua própria cor (a dinamica de TextColor não vai fazer nada) Quando o valor é de 1 a 9, o objeto terá a cor vermelha. Quando o valor é superior a 10, o objecto terá a cor azul.

Dinâmica Bargraph

Tabela 10: Configuração da Dinâmica Bargraph

Configuração	Descrição
Bargraph	Dinâmica de gráfico de barras
Expression	o valor usado para a dinâmica de gráfico de barras.
Value Range	Os valores mínimos e máximos que correspondem ao percentual de preenchimento máximo e mínimo.
Fill (%)	O percentual mínimo e máximo de preenchimento. da barra gráfica

Tabela 10: Configuração da Dinâmica Bargraph

Configuração	Descrição
Bar Color	A cor do gráfico de barras.
Orientation	A orientação do gráfico de barras: para cima centro horizontal para baixo para a direita centro vertical para a esquerda

Dinâmica Visibility

Tabela 11: Configuração da Dinâmica Visibility

Configuração	Descrição
Visibility	Altera dinamicamente a visibilidade e opacidade do objeto
Visible	Digite um tag, uma propriedade, ou uma expressão que retorne um valor. Considerando-se o valor resultante: • Zero - Objeto será visível. • Maior que zero - Objeto serão ocultado. Para mais informações, consulte "Configurando Expressões" na pagina 141.
Tooltip	Um texto que será exibido como uma dica de ferramenta
Opacity	Opacidade • ObjectValue - O valor usado para definir a opacidade. • Range - Os valores mínimos e máximos que correspondem ao máximo e mínimo. de opacidade. • Opacidade - A mínima e máxima opacidade (0 - invisível, 0,5 - um pouco transparente, 1 - opaco).

Dinâmica MoveDrag

Tabela 12: Configuração da Dinâmica MoveDrag

Configuração	Descrição
MoveDrag	Movimenta dinamicamente o objeto
BindingMode	• TwoWay - Entrada e saída causam movimentação • InputOnly - Apenas entrada causa movimento. O objeto não se move quando o valor do objeto muda. • OutputOnly - Apenas saída causa movimento. O objeto não se move com interação do usuário.
Horizontal Move	• Expressão com a referência para movimento horizontal • Range - Os valores mínimos e máximos que correspondem à posições horizontais mínima e máxima. • Posição - As posições horizontais mínima e máxima
Vertical Move	• Expressão com a referência para movimento vertical • Range - Os valores mínimos e máximos que correspondem à posições verticais mínima e máxima. • Posição - As posições verticais mínima e máxima

Dinâmica Scale

Tabela 13: Configuração da Dinâmica Scale

Setting	Descrição
Scale	Muda o tamanho do objeto dinamicamente.
Width Scale	• Expressão - o valor usado para a escala de largura. • Range - Os valores mínimos e máximos que correspondem aos percentuais minimo e máximo da largura de escala • Escala (%) - O percentual mínimo e máximo de largura de escala.
Height Scale	• Expressão - o valor usado para a escala de altura • Range - Os valores mínimos e máximos que correspondem aos percentuais minimo e máximo da altura de escala • Escala (%) - O percentual mínimo e máximo de altura de escala.
Scale Reference:	Centro Esquerda Para cima Direita Para baixo

Dinâmica Rotate

Tabela 14: Configuração da Dinâmica Rotate

Configuração	Descrição
Rotate	Gira o objeto dinamicamente.
Expression	o valor utilizado para a rotação.
Value Range	Os valores mínimos e máximos que correspondem ao ângulo máximo e mínimo de rotação. Por exemplo: de 0 a 100
Angle	O ângulo mínimo e máximo de rotação. Por exemplo: 0-360
Center Reference	Centro Esquerda Para cima Direita Para baixo

Dinâmica Skew

Tabela 15: Configuraçõa da Dinâmica Skew

Configuração	Descrição
Skew	Distorce o objeto dinamicamente.
X-axis Skew	• ObjectValue - o valor usado para a inclinação do eixo-X. • Range - Os valores máximo e mínimo que corresponde ao ângulos de desvio máximo e mínimo do eixo X. Por exemplo: de 0 a 100. • Skew (º) - O mínimo e o máximo ângulo de desvio do eixo X. Por exemplo: de 0 a 180 º.

Tabela 15: Configuraçõa da Dinâmica Skew

Configuração	Descrição
Y axis skew	• ObjectValue - o valor usado para a inclinação do eixo-Y. • Range - Os valores máximo e mínimo que corresponde ao ângulos de desvio máximo e mínimo do eixo Y. Por exemplo: de 0 a 100. • Skew (º) - O mínimo e o máximo ângulo de desvio do eixo Y. Por exemplo: de 0 a 180 º.
Skew Reference	Centro Esquerda Para cima Direita Para baixo

Dinâmica TextOutput

Tabela 16: Configuração da Dinâmica TextOuput

Configuração	Descrição
TextOutput	Dinâmica de saída de texto.
Expression	Indica o texto que será mostrado no objeto.
Localizable	Indica se o texto deve ser traduzido quando o dicionário muda.
DesignModeCaption	O valor mostrado no modo de design: • ShowObjectNames-O conteúdo do campo de texto é mostrado exatamente como ele é. • ShowPlaceHolders - São mostrados os caracteres # # #, o número de caracteres é definido pelo campo
MaxLength	Define o número máximo de caracteres.

Controles da interface do usuário

- Para configurar a dinâmica, consulte "Dinâmicas e Animações" na pagina 166
- Para configurar um painel circular, "Controles da interface do usuário" na pagina 176.
- Para configurar um gráfico de pizza, "Configurando um Gráfico de Pizza" na pagina 177
- Para configurar um gráfico de barras, consulte "Configurando um Gráfico de Barras (BarChart)" na pagina 177.
- Para configurar uma janela de alarme, consulte "Configurando uma Janela de Alarmes" na pagina 178

- Para configurar uma janela de tendência, consulte "Configurando a Janela de Tendência" na pagina 179
- Para configurar um DataGridWindow, consulte "Configurando uma Janela DataGrid" na pagina 181.

Configurando um painel circular

Dê um duplo clique em um objeto painel circular na tela para configurá-lo. a Tabela 17 descreve a configuração do painel Circular.

Tabela 17: Configuração de Painel Circular

Campo	Descrição
Control Name	Define um nome para o controle, desta forma ele poderá ser acessado no script CodeBehind. Ver "Code Behind - Scripts de Telas" na pagina 160.
Columns	Para cada valor que você deseja incluir, selecione uma seção e configure as definições, à direita da lista , que estão descritas a seguir. O painel exibe uma seção para cada coluna.
Image	Escolha em Resources uma imagem para exibir.
Preview	Somente leitura. Exibição de imagem selecionada.

Configurando um Gráfico de Pizza

Dê um duplo clique em um objeto PieChart na tela para configurá-lo. A Tabela 18 descreve a configuração de gráfico de pizza.

Tabela 18: Configuração de gráfico de Pizza

Campo	Descrição
Control Name	Define um nome para o controle, então ele pode ser acessado no script CodeBehind. Ver "Code Behind - Scripts de Telas" na pagina 160.
ChartType	Selecione o tipo de gráfico.
Data Items	Para cada valor que você deseja incluir, selecione uma seção e configure as definições, ao lado direito da lista, que estão descritas a seguir.. O gráfico mostra uma seção para cada item de dados..
FieldName	Digite um nome para o item de dados.
LinkedValue	Digite um nome de tag como fonte de dados para o item de dados.
Brush	Clique para selecionar a cor para o item de dados.

Configurando um Gráfico de Barras (BarChart)

Dê um duplo clique em um objeto BarChart na tela para configurá-lo. A Tabela 19 descreve a configuração de gráfico de barras.

Tabela 19: Configuração de Gráfico de Barras

Campo	Descrição
Control Name	Defina um nome para o controle, para que ele possa ser acessado no script CodeBehind. Ver "Code Behind - Scripts de Telas" na pagina 160
Type	Selecione o tipo de gráfico.
Data Source	Entre o nome do Dataset Table ou query para usar para o gráfico.
Grid Lines	Clique para selecionar a cor das linhas de grade.
Window	Clique para selecionar a cor para o fundo da janela com o gráfico.
Labels	Clique para selecionar a cor para os rótulos.
Show Horizontal Labels 45°	Selecione para mostrar os rótulos do eixo horizontal em angulo de 45°.
Show value over bar	Selecione para exibir o valor da barra acima da barra.
Data Items	Para cada coluna, existente na fonte de dados, que você deseja incluir, marque uma barra e configure as definições ao lado direito da lista, que estão descritas a seguir. O gráfico exibe uma barra para cada linha de definição.
FieldValue	Nome da coluna no banco de dados.
Min.	Digite uma tag para definir o valor mínimo.
Max.	Digite uma tag para definir o valor máximo
Brush	Clique para selecionar a cor para o item de dados.

Configurando uma Janela de Alarmes

Dê um duplo clique em um objeto Janela de Alarme em uma tela para configurá-lo. A Tabela 20 descreve a configuração da janela de alarme.

Tabela 20: Configuração da Janela de Alarmes

Campo	Descrição
Control Name	Defina um nome para o controle, para que ele possa ser acessado no script do CodeBehind. Ver "Code Behind - Scripts de Telas" na pagina 160..
MaxLines	O número máximo de linhas exibidos na janela.
List	Selecione o tipo de alarmes que serão exibidos na janela: • OnlineAlarms - Exibe apenas os alarmes ativos. Os Alarmes são exibidos somente enquanto o item de alarme está no estado de alarme. • AlarmHistory - Exibe somente alarmes passados. • Events - Mostra apenas os SystemEvents (Edit> Alarms> Groups). • AlarmHistory + Events - Exibe alarmes passados e SystemEvents
Merge Hi and HiHi Lines	Combinar alarmes configurados como Hi e HiHi em uma única linha.

Tabela 20: Configuração da Janela de Alarmes

Campo	Descrição
History Interval/ To	Introduza as datas de início e fim do intervalo para o qual os alarmes devem ser incluídos.
Show Column Titles	Selecione para exibir os títulos das colunas.
Ack by Page	Digite um nome de tag, que quando disparado, reconhece os alarmes na janela de alarmes.
Show Group Control, Label	Digite o texto a ser exibido acima dos títulos das colunas.
AllowSort	Selecione para permitir que usuários classifiquem os dados por coluna.
Allow Column Reorder	Selecione para permitir que os usuários alterem a ordem das colunas.
Filter	Digite uma consulta SQL para limitar os alarmes exibidos.
Refresh	Digite um nome de tag que irá desencadear uma atualização dos dados de alarme.
Display Millisecond	Selecione para incluir milissegundos na hora do alarme.
Ack Selected Line	Configure definições para reconhecer um alarme.
Columns	Para cada coluna na lista de colunas , selecione a coluna e configure as definições ao lado direito da lista, que estão descritas a seguir
Visible	Selecione esta opção para a coluna ser exibida na janela de alarme.
Allow Filter	Selecione a opção para que os usuários possam filtrar nesta coluna
Show in Column Chooser	Selecione para incluir a coluna no seletor de colunas, o que permite aos usuários selecionar, em tempo de execução as colunas a serem mostradas
Title	Digite o texto para o título da coluna.
Width	Digite a largura da coluna inicial, em unidades do WPF (pixels independentes de monitor).
Sort	Escolha como ordenar inicialmente por esta coluna.

Configurando a Janela de Tendência

Dê um duplo clique em um objeto Janela de Tendência na tela para configurá-lo. A Tabela 21 descreve a configuração da janela de tendência.

Tabela 21: Configuração da Janela Tendência

Campo	Descrição
Control Name	Defina um nome para o controle, para que ele possa ser acessado no script do CodeBehind. Ver "Code Behind - Scripts de Telas" na pagina 160..
0(online) 1(history)	0 - o controle mostra os dados em tempo real (on-line.) 1 - o controle mostra os dados do histórico. Por exemplo: Tag.onLineHist.

Tabela 21: Configuração da Janela Tendência

Campo	Descrição
OnlineTrigger	A taxa de atualização do controle de tendência online. Por exemplo, 00:00:01 ou {Tag.trendTimeSpan}.
HistoryDateTime	O ponto inicial para o controle histórico tendência. Por exemplo, {Tag.initialTrendHistory}.
Max Samples	Define o número máximo de amostras que serão coletadas do banco de dados.
Window	Define a cor de fundo do controle de tendência.
Labels	Definir a cor dos rótulos do controle de tendência
Cursor Enable	Ativar (marcado) ou desativar (desmarcado) a exibição do cursor vertical.
CursorPosition (%)	Indica a posição do cursor, onde 0 significa posição inicial e 100 significa posição final. Por exemplo: Tag.cursorPos.
Cursor Output	Indica o valor do eixo X para a posição atual do cursor. Por exemplo, Tag.cursorOut.
Cursor Color	Define a cor do cursor.
Pens Legend	Define a posição do quadro de legenda das penas no controlo de tendência.
Y Axis	
Range	Define os valores mínimo e máximo para o eixo Y.
Labels	Define a quantidade de linhas de grade horizontais.
Format	O formato dos valores do eixo Y. Para os formatos numéricos válidos, consulte "Formatos de Tags" na pagina 84. Por exemplo, N1 (número com uma casa decimal).
X Axis	
Interval	Define o intervalo de tempo do eixo-X.
Labels	Define a quantidade de linhas de grade verticais
Format	O formato do eixo X é definido por dois campos: formato da primeira linha e formato de segunda linha. Isto é especialmente útil para definir etiquetas para marcas que requerem duas linhas de informação. Para formatos válidos de data e hora consulte "Formatos de Tags" na pagina 84. Por exemplo, t (tempo) para o formato da primeira linha, d (data abreviada) para o formato da segunda linha.
Pens	
Visible	Mostra (1) ou oculta (0) a pena selecionada. Ex 1 ou {showPen1}.
Tag	Define a Tag que irá fornecer o valor para a pena
Min	Minimo valor da escala linear para o valor da tag, de acordo com o intervalo do eixo Y.
Max	Máximo valor da escala linear para o valor da tag, de acordo com o intervalo do eixo X.
Pen	Escolhe o modelo, a cor, e a espessura da linha da caneta.

Tabela 21: Configuração da Janela Tendência

Campo	Descrição
Mark	Define o formato da marca para cada ponto da linha de tendência.
CursorValue	Definir a tag que vai receber o valor real do eixo Y, de acordo com a posição do cursor. Por exemplo: Tag.pen1CursorValue.
FieldName	Define o nome da pena.

Usando a escala no eixo Y

Considerando-se:

Eixo Y Min = 0;

Eixo Y Max = 100;

O controle de tendência permite que muitas penas sejam exibidas juntas. Quando suas penas não tem a mesma escala, você pode usar as dicas abaixo para ajustar seus dados no mesmo gráfico, para melhor visualização:

- Se uma pena tem uma escala mais baixa, de 0 a 1, por exemplo, você pode definir a propriedade Max da pena em 1, assim quando o valor real do Tag for 1, o valor 100 será exibido no gráfico. (escala 100/1).
- Se uma pena tem uma escala maior, de 0 a 1000, por exemplo, você pode definir a propriedade Max da pena para 1000, assim quando o valor real do Tag for 1000, o valor 100 será exibido no gráfico. (Escala 1/10).

Configurando uma Janela DataGrid

Dê um duplo clique em um objeto janela DataGrid sobre uma tela para configurá-lo. A Tabela 22 descreve a configuração da janela DataGrid. Campos críticos que você deve configurar são mostrados com um *.

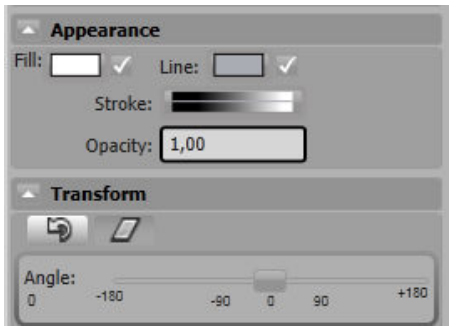
Tabela 22: Configuração de uma janela DataGrid

Campo	Descrição
Control Name	Defina um nome para o controle, para que ele possa ser acessado no script do CodeBehind. Ver "Code Behind - Scripts de Telas" na pagina 160.
Data Source	Entre o nome do Dataset Table ou query para usar na janela
SelectedValues	Digite um tag ou matriz de tags para receber o conteúdo da linha selecionada.
SelectedIndex	Digite um tag para receber o número da linha atualmente selecionada. A numeração de linhas começa com 0.
LinesCount	Selecione um tag para receber o número de linhas na janela.
Theme	Selecione o tema para a janela.
BindindMode	Escolha o modo de ligação entre dados e tabela que deva ser utilizado. Veja em http://msdn.microsoft.com/en-us/library/system.windows.data.bindingmode.aspx.
AllowInsert	Selecione para que os usuários possam adicionar linhas.

Tabela 22: Configuração de uma janela DataGrid

Campo	Descrição
Show Group Control, Label	Digite o texto a ser exibido nos títulos das colunas.
Show Column Titles	Selecione para exibir os títulos das colunas.
AllowSort	Selecione para permitir que usuários classifiquem os dados por coluna.
Auto Create Column	• Selecione para que o sistema automaticamente inclua na janela todas as colunas da tabela fonte. Se selecionado, você não tem que configurar todas as colunas com as opções abaixo.. Para personalizar colunas específicas, adicione a coluna na lista de colunas e defina as configurações do lado direito, descritas abaixo. • Desmarque esta opção para configurar manualmente cada coluna que você deseja incluir na lista usando as configurações do lado direito, descritas abaixo.
Filter	Digite uma consulta SQL para limitar os dados exibidos.
Refresh	Digite um nome de Tag que irá disparar uma atualização dos dados.
Columns	Para personalizar colunas específicas, adicione a coluna na lista de colunas e defina as configurações do lado direito, descritas abaixo.
Visible	Selecione para que a coluna seja exibida na janela do DataGrid
Editable	Selecione para permitir que os usuários edite a coluna.
Show in Column Chooser	Selecione para incluir a coluna no seletor de colunas, o que permite aos usuários selecionar as colunas a serem mostradas.
FieldName	Digite um nome para a coluna.
FieldType	Selecione o tipo de dados da coluna.
Title	Digite um texto para o cabeçalho da coluna.
Width	Digite a largura da coluna inicial, em unidades do WPF (pixels independentes de monitor).
Sort	Escolha como ordenar inicialmente a coluna.

Aparência e Transformação



Estas definições podem ser feitas para controles sobre uma tela. Definem a aparência de um controle e uma maneira de girar ou de distorcer.

Appearance

Fill - Ao clicar no quadrinho, aparece uma janela de escolha do pincel para preenchimento do controle, que inclui, cores, gradientes e textura.

Line - Aparece a mesma janela, acima descrita, para a escolha da linha de contorno do controle.

Stroke - Clicando na figura, aparece uma janela para escolha do "Stroke", isto é formato da linha, po cheia, pontilhada, tracejada, etc.

Opacity - Determina a opacidade do fundo do controle, numero fracionário entre 0 e 1. 0 é transparente e 1 é opaco.

Transform

A seta define rotação do controle. Escolha o angulo de rotação na régua abaixo.

O ícone define distorção do controle. Escolha ângulos de distorção nos eixos X e Y do plano da tela

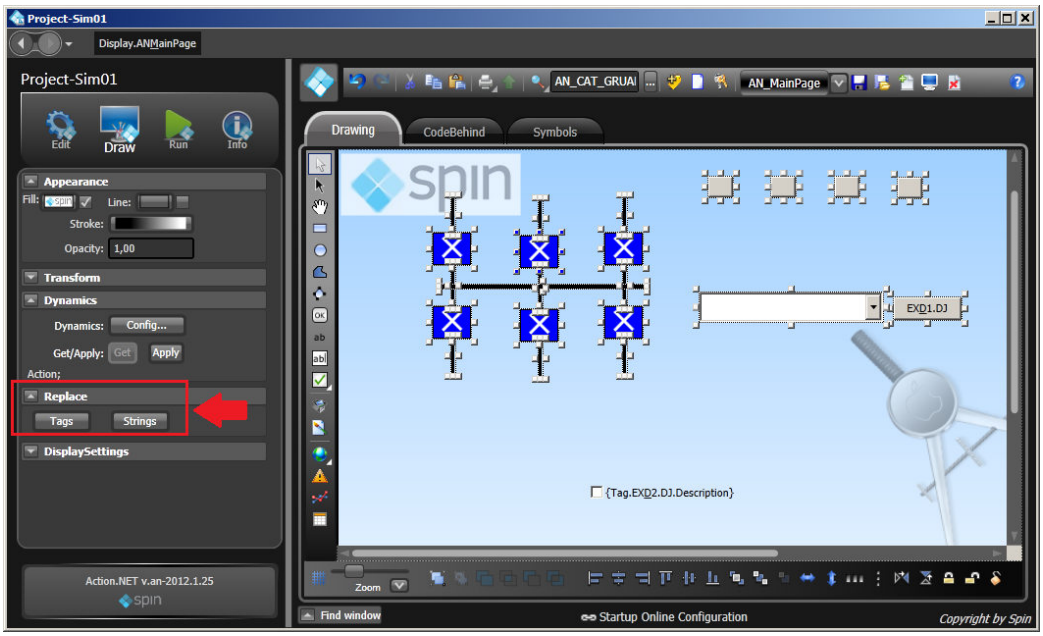
---

Alteração de vários controles

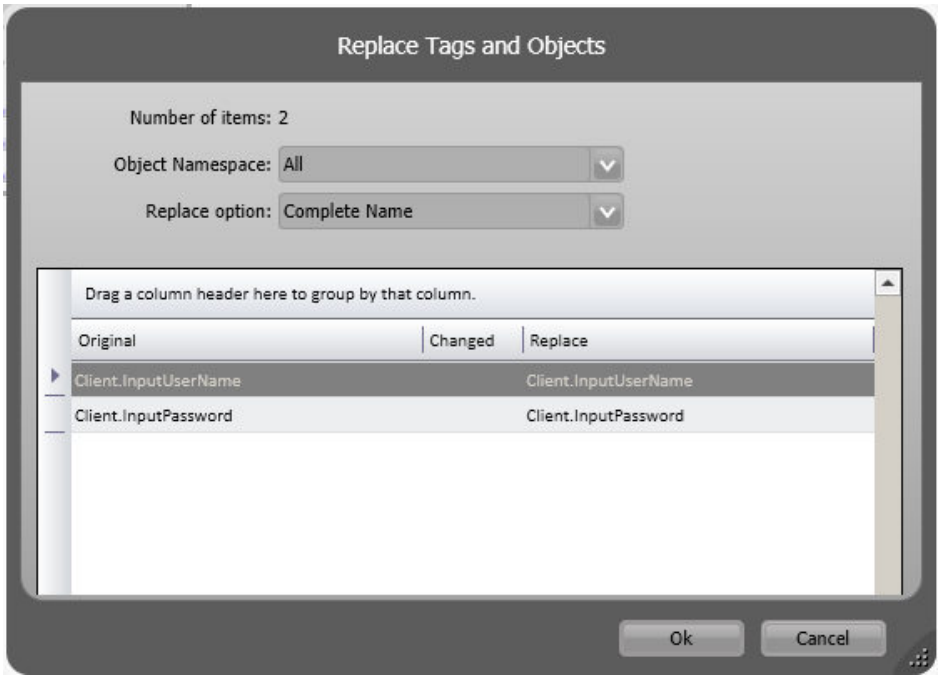
A funcionalidade de **Replace** é oferecida quando se seleciona vários controles em uma Tela.

Se você deseja fazer a alteração de um conjunto de *tags* ou *textos* existentes em uma tela, Selecione todos eles e utilize a funcionalidade **Replace**, conforme mostrado nas duas figuras abaixo.

1. A figura abaixo mostra seleção de diversos objetos:



2. Ao pressionar a opção Tags, são mostrados os tags usados nestes controles. Se alterarmos os tags da direita, eles serão sobrepostos na tela pelos seus correspondentes da esquerda.



- No caso de textos, todos os Strings usados nos objetos selecionados serão apresentados em forma semelhante à apresentada acima, para serem alterados.

Gerenciando Imagens e Símbolos

Para utilizar imagens em suas telas ou relatórios você pode importá-las através da aba Recursos no ambiente Displays.

Já vem incorporada ao Action.NET uma biblioteca de imagens e símbolos. Você pode criar novos e pode importar de um projeto para outro, criando assim um repositório de imagens e símbolos.

Nesta seção são descritas estas possibilidades e a maneira de usá-las:

- "Recursos de imagens" na pagina 184
- "Gerenciando Símbolos" na pagina 185

Recursos de imagens

Você deve importar todas as imagens que você deseja usar em suas telas ou relatórios através da aba **Recursos** no ambiente **Displays.** Isso cria um repositório de imagens para a aplicação.

Se você precisar atualizar uma imagem com uma nova versão, você pode substituí-la, e ele vai ser atualizada em todos os locais da aplicação em que estiver sendo utilizada. Certifique-se de manter o mesmo nome do recurso, quando fizer troca de imagens.

Você pode usar qualquer arquivo de imagem para preencher ou pintar um objeto ao criar telas em **Draw> Drawing** .

Você pode importar os seguintes tipos de arquivo:

- . bmp
- . gif
- . ico
- . jpg
- . png
- . tiff
- . wdp

Para gerenciar recursos:

1. Vá para **Edit> Displays> Resources**.
2. Clique **Import Images**.
3. Navegue até onde estão as imagens que você deseja importar.
4. Selecione os arquivos de imagem e clique em **Open**.
5. Em cada linha, digite ou selecione as informações, conforme necessário.

Coluna	Descrição
Name	Edite o nome padrão para um nome descritivo.
Description	Digite uma descrição desta imagem.

Gerenciando Símbolos

A biblioteca de símbolos vem com uma série de gráficos que você pode usar. Você também pode criar símbolos para adicionar à biblioteca. Os símbolos podem ser apenas um gráfico ou um gráfico e a configuração de uma dinâmica.

Para visualizar símbolos:

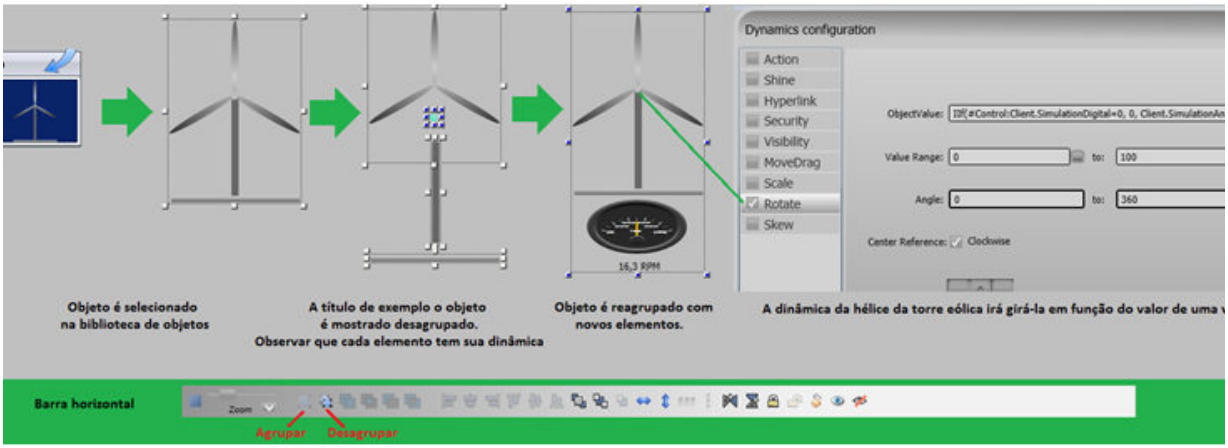
1. Siga uma das seguintes opções:
   - Para ver apenas os símbolos utilizados neste projeto, vá para **Draw > Symbols**.
   - Para ver todos os símbolos disponíveis, vá para **Draw > Drawing** e clique em **Open Symbol Library**.
2. Para salvar um símbolo como um arquivo de imagem, selecione a linha e clique em **Save to Image**.

Para adicionar um símbolo

1. Vá para **Draw > Drawing.**
2. Desenhe o objeto que você deseja tornar-se um símbolo.
3. Selecione todo o objeto e clique com o botão direito.
4. Selecione **Make New Symbol**.

Organizando a Biblioteca de símbolos

Usando as ferramentas de desenvolvimento gráfico pode-se criar novos símbolos para utilizar em projetos. A biblioteca já vem com um conjunto de símbolos tais como Chaves, Tanques, Medidores e Bombas, Botões, Gauges, Sliders e outros. Estes novos objetos são agrupamentos de objetos existentes com todas as dinâmicas associadas. Abaixo, a título de exemplo, é mostrado o objeto *Fan* representando uma eólica. Este objeto é desagrupado e após reagrupado adicionando-se ao existente um tacômetro e um valor indicando a velocidade em RPM. Observar que na dinâmica da hélice, está selecionado rodá-la quando um valor for maior que 1 (ligado).



Os símbolos são objetos complexos criados usando as ferramentas de edição de telas do Action.NET. Assim, por exemplo, um objeto disjuntor deve ser uma figura com todas as propriedades e métodos associados a um disjuntor em uma tela de um unifilar.

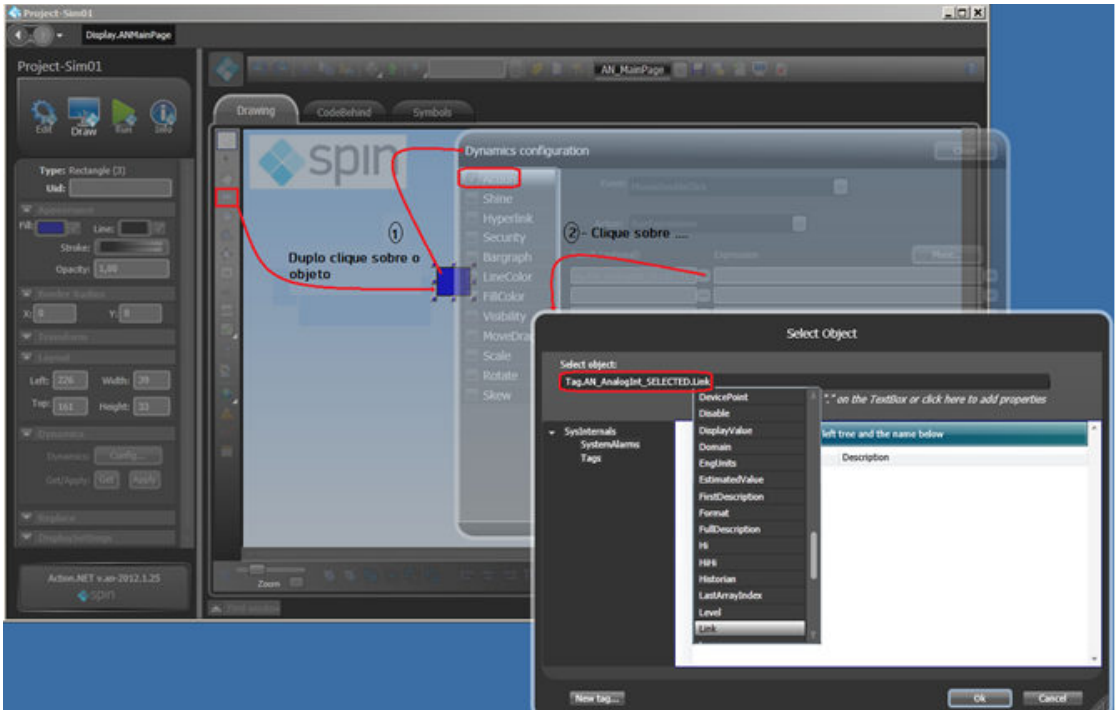
Cada símbolo deve ter um nome que o identifica, uma categoria que o agrupa entre seus semelhantes e uma descrição que auxilia no seu uso por técnicos que deverão configurar as telas do SCADA.

No caso de uma empresa integradora que trabalhe com diversas concessionárias de energia, uma possibilidade de organização é agrupar todos os objetos (disjuntores, seccionadores, medidas analógicas, etc.) associados à cultura da concessionária a uma categoria, como o exemplo CEB_SIMBOLOS (Concessionária de distribuição de Brasília).

Um objeto disjuntor terá, por exemplo, as seguintes dinâmicas:

1. Quando o mouse passa sobre ele, ele aumentará de tamanho;
2. Ele será um quadrado com quatro cores distintas em função de estar: Bloqueado (0), Aberto (1), Fechado (2) e Indefinido (3);
3. Um duplo clique sobre ele deverá abrir uma janela de comando que permitirá ligá-lo, desligá-lo, impedi-lo ou desativá-lo.

Criando um Símbolo (Disjuntor)

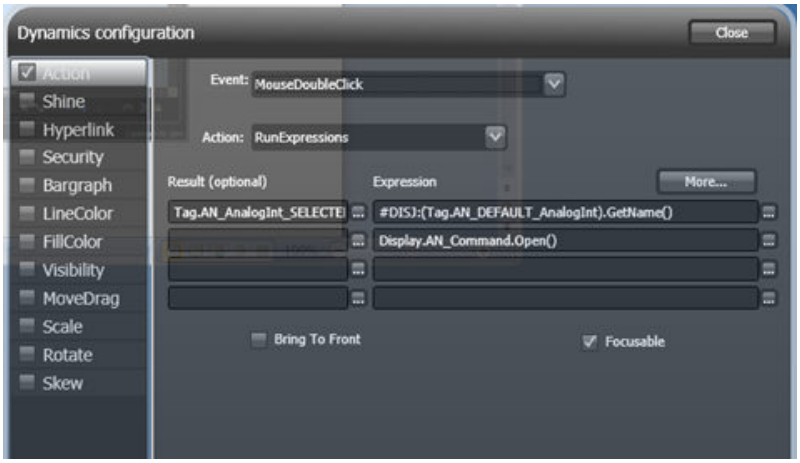


Por exemplo para criar um símbolo para o objeto Disjuntor:

1. Sobre a aplicação Default, é criado um objeto retângulo, no formato de um disjuntor. Um duplo clique sobre este objeto abre a janela de dinâmicas para configuração.
2. Selecionamos a dinâmica
3. **Action** e atribuímos as seguintes ações:

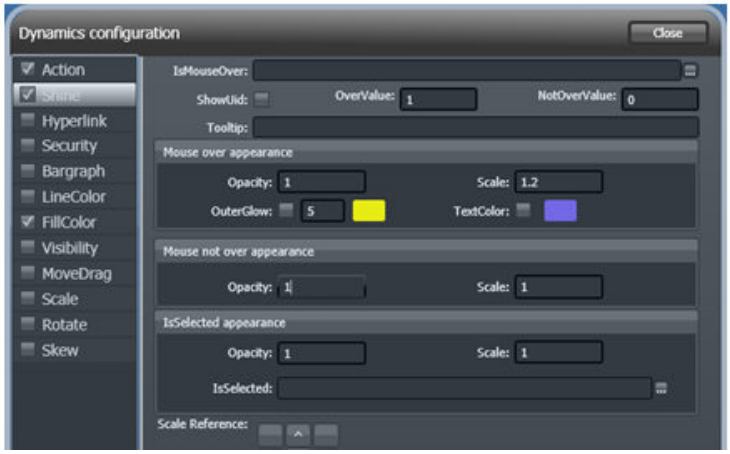
```
Tag.AN_AnalogInt_SELECTED.Link =
#DISJ:(tag.AN_DEFAULT_AnalogInt).GetName()
```

- A primeira ação será jogar o link (reference) da variável associada ao Label `#DISJ` na variável à esquerda: `AN_AnalogInt_SELECTED`. Na construção do símbolo usamos uma variável Dummy (AN_DEFAULT_AnalogInt) que durante o projeto das telas deverá ser substituída pala variável que descreve o estado do disjuntor.
- `Display.AN_Command.Open()`: A segunda ação será chamar a janela AN_Command que deverá ter como parâmetro de entrada a variável `globalTag.AN_AnalogInt_SELECTED` , já que na primeira linha da dinâmica ponteiro desta variável aponta para a variável dummy que representa o estado do disjuntor.
- Abaixo é apresentada a dinâmica **ação** completa:

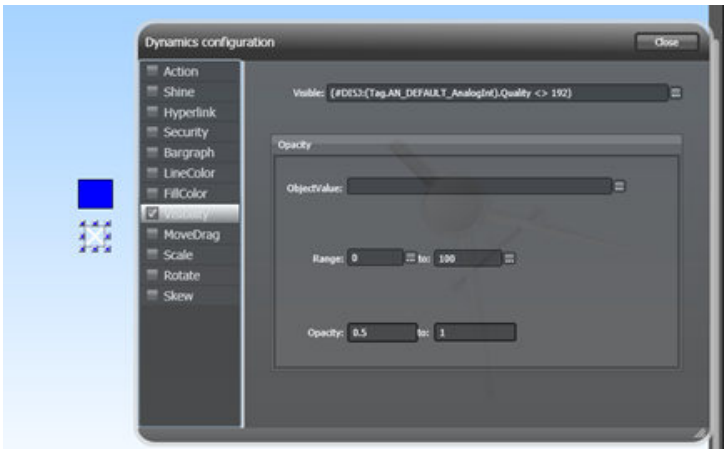


4. A segunda dinâmica será a FillColor e usaremos o valor da variável associada `(tag.AN_DEFAULT_AnalogInt).Value` para atribuir as cores (0) Branco, (1) Verde, (2) Vermelho e (3) Amarelo.

5. A última dinâmica associada a este objeto retângulo será Shine que fará que o objeto aumente em 20% (1,2) quando o mouse passar sobre ele.



6. Outra ação implementada será colocar um "X" sobre o objeto caso não exista comunicação com o campo. Para isso, cria-se uma figura "X" (Duas retas agrupadas) e define-se que ele será invisível quando a qualidade tag for diferente de 192 (qualidade ok).



7. Finalmente, coloca-se um objeto sobre o outro, agrupa-se, e cria-se o símbolo disjuntor.

Janela de Comando do Disjuntor

A janela de comando do disjuntor é apresentada abaixo. Ela manipula a variável global: `AN_AnalogInt_SELECTED` e possui os cabeçalhos localizáveis:

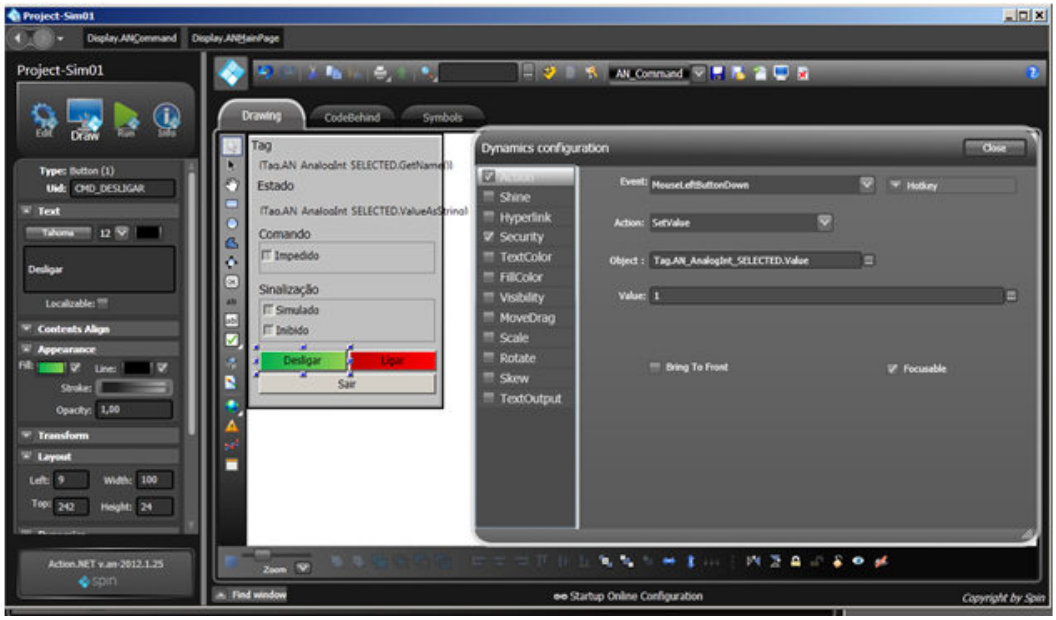
- Tag: apresenta o nome da variável associada ao disjuntor:

  `Tag.AN_AnalogInt_SELECTED.GetName()`
- Estado: apresenta o estado na variável associada ao disjuntor:

  `Tag.AN_AnalogInt_SELECTED.ValueAsString`
- Comando: Indica se o comando está ou não impedido (no caso apenas a check-box)
- Sinalização: também check-box de estados;

Além destes parâmetros de "labels" existem três botões, onde a cada botão está associada uma dinâmica:

- Desligar: `Tag.AN_AnalogInt_SELECTED.Value = 1`
- Ligar: `Tag.AN_AnalogInt_SELECTED.Value = 1`
- Sair: Ação CloseDisplay.

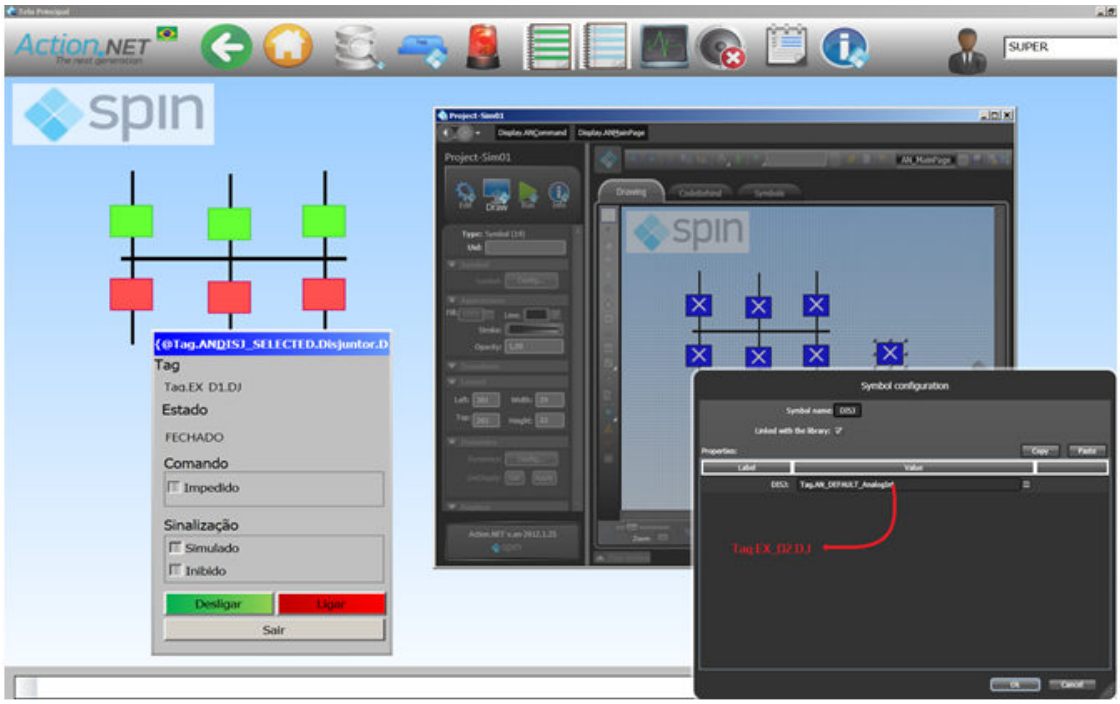


Criando um Unifilar com os Símbolos Disjuntores

Na figura a seguir, foi criado um unifilar com uma barra e seis símbolos DISJ. Para cada símbolo, como pode ser observado na figura, substituiu-se a variável `Tag.AN.DEFAULT.AnalogInt` por uma variável associada a um disjuntor.

O comando ligar / desligar muda o valor do Tag (1=desligar / 2=Ligar) e sua figura associada.

Da mesma forma como se criou este objeto disjuntor, pode-se criar todos os objetos associados à cultura do cliente.



Operações Usuais na Criação de novos Símbolos

É comum ao se gerar novos itens na biblioteca de Símbolos (Symbols):

1. Associar Labels a variáveis do tipo dummy:

```
#LAB1:(Tag.AN_AnalogInt_Selected)
```

2. Usar Tags tipo reference para apontar para as variáveis dummy como em:

```
 [Tag.AN_AnalogInt_Selected.Link
=#LAB1:(Tag.AN.Default.AnalogInt).GetName()]
```

3. Fazer todas as operações com a variável reference, após fazer seu apontamento

Para criar uma variável dummy que assumirá o papel de uma variável de uma rotina associada a um símbolo, por exemplo: Disjuntor, crie uma variável dummy do mesmo tipo da variável que contém o estado do disjuntor e crie uma variável do tipo reference.

Por exemplo, se temos um disjuntor que é um AnalogInt, criamos uma variável dummy do mesmo tipo como, por exemplo, `Tag.AN.Default.AnalogInt` e criamos uma variável global do tipo reference como, por exemplo, `tag.AN_AnalogInt_Selected.`

Depois de apontar a variável tipo Reference para o disjuntor: `[Tag.AN_AnalogInt_Selected.Link =#LAB1:(Tag. AN.Default.AnalogInt).GetName()]`,

os seguintes atributos poderão ser usados:

- Estado do disjuntor: `(Tag.AN_AnalogInt_Selected).value (0=bloqueado, 1=aberto, 2=fechado, 3=Indefinido);`
- Qualidade da leitura do estado do disjuntor: `(Tag.AN_AnalogInt_Selected).quality`
- Descrição do disjuntor: `(Tag.AN_AnalogInt_Selected).description`

- Disjuntor Inibido (alarme desabilitado): `(Tag.AN_AnalogInt_Selected).AlarmDisable = 1`
- Disjuntor Simulado: `(Tag.AN_AnalogInt_Selected).locked = 1`
- Disjuntor alterou o valor: `(Tag.AN_AnalogInt_Selected).changed`
- Estado do disjuntor em texto: `(Tag.AN_AnalogInt_Selected).ValueAsString` – Neste caso deve existir um dicionário associado;
- String com Tag do disjuntor: `(Tag.AN_AnalogInt_Selected).GetName()`

Objetos runtime Display

O espaço de nomes **`Display`** lista todas as Telas com suas propriedades e métodos de abrir e fechar.

O namespace **`Layout`** lista todos os layouts com suas propriedades e métodos de abrir e fechar.

O namespace **`Client`** tem as propriedades do ambiente em cada computador cliente ou dispositivo móvel conectado.

Veja http://www.spinengenharia.com.br/help/an-2014/runtime/index.html para a referência de programação completa em objetos de tempo de execução.

Relatórios e Acesso a dados

As seções seguintes descrevem como gerar relatórios:

- "Criando Relatórios" na pagina 193
- "Usando o Editor de Relatórios" na pagina 194
- "Objeto runtime Report" na pagina 194

Criando Relatórios

O ambiente **Reports** permite configurar o formato de relatórios e sua formatação básica. Os Usuários da aplicação podem então executar os relatórios assim configurados.

Para criar relatórios:

1. Vá para **Edit > Reports > Reports**.
2. Digite ou selecione as informações, conforme necessário.

Coluna	Descrição
Name	Digite um nome para o relatório. O sistema permite que você saiba se o nome não é válido.
Padding	Selecione o preenchimento a ser usado na substituição de um nome de Tag com o seu valor (o campo é criado com o número de caracteres do nome do Tag): • Compact - Remove todos os caracteres extras e exibe apenas o valor do tag. • PadRight - coloca espaço extra para cada caractere à direita do valor tag. • PadLeft - coloca espaço extra para cada caractere à esquerda do valor tag.
SaveFormat	Selecione o formato do relatório • XPS • HTML • Unicode • ASCII • PDF
SaveFileName	Digite um texto, para o nome do arquivo com {ObjectProperties}. Use o caminho completo.
Description	Digite uma descrição do relatório
[Outras colunas]	Para definições de outras colunas que estão disponíveis em muitas tabelas, consulte "Definição de colunas comuns" na pagina 68.

Usando o Editor de Relatórios

Você cria o texto, faz a formatação e define valores para o relatório usando o editor de textos.

Para usar o editor de texto:

1. Vá para **Edit > Reports > TextEditor**.
2. Crie o texto, tags e formatação do relatório.
3. Clique em **Save**.
4. Configure as propriedades do relatório para a dinâmica de exibição ou script que irá executar o relatório.

Objeto runtime Report

O namespace `Report` lista todos os relatórios configurados.

Os métodos mais comumente utilizados do Report é o comando Salvar:

Report.Report1.SaveCommand()

Veja http://www.spinengenharia.com.br/help/an-2014/runtime/index.html para a referência de programação completa em objetos de tempo de execução

Execução, Testes e Publicação de Projetos

As seções seguintes descrevem como executar, "build" e testar aplicativos:

- "Executando a Aplicação" na pagina 195
- "Construindo e Empacotando Projetos" na pagina 196
- "Controle de Versão de Projetos" na pagina 199
- "Objetos runtime Information" na pagina 201

Executando a Aplicação

Você pode testar a aplicação de uma das seguintes formas:

- Usando **Run> Test** - Permite que você execute o projeto em modo de segurança, ativando o aplicativo com algumas características protegidas. Os dados históricos, alarmes e eventos são gravados para arquivos temporários para que não afetem os registos de produção, e os módulos de comunicação não podem gravar dados nos IEDs.
- Usando **Run> Startup** - Permite você executar o projeto em pleno funcionamento, sem a necessitar o "Build" do projeto.

Você pode executar o projeto usando ambos **Test** e **StartUp** ao mesmo tempo.

Para obter informações sobre como utilizar as ferramentas de diagnóstico disponíveis, consulte "Ferramentas para diagnose" na pagina 112

Para testar o aplicativo:

1. Siga uma das seguintes opções:
   - Vá para **Run> Test**.
   - Vá para **Run> Startup**.
2. Digite ou selecione as informações, conforme necessário.

Campo	Descrição
UserName	Digite um nome de usuário válido para acessar o aplicativo.
Password	Digite a senha que corresponde ao nome do usuário.
Project Server	Somente leitura. Exibe o endereço IP ou nome do computador onde o projeto está baseado conforme a configuração na aba **Server**. Veja "Testando e executando projetos" na pagina 40.
Port	Somente leitura. Exibe a porta usada pelo aplicativo para o acesso. Para o Teste, usa 3201. Para o Startup, ele usa 3101. Estas portas devem ser abertas no servidor.
Startup Computer	Somente leitura. Mostra se o servidor configurado esta no computador local ou em um computador remoto.

Campo	Descrição
Execution Path	Permite mudar o caminho de execução padrão, que é o local do arquivo do projeto.
Module Information	Execute a ferramenta Módulo de Informação.
Property Watch	Execute a ferramenta Proprierty Watch.
Trace Window	Execute a ferramenta TraceWindow.
Run Modules	Selecione quais Módulos executar durante a execução do projeto.
Status	O campo mostra o status atual em execução e o status conectado ou desconectado. • Clique em **Try to Connect** para se conectar ao aplicativo em execução. Necessário para que as alterações feitas ao projeto sejam aplicadas imediatamente para o execução de teste ou de Startup. • Clique em **Disconnect** se você não quiser que as alterações feitas no projeto sejam aplicadas imediatamente para a execução em modo Test ou Startup • Clique em **Stop** para interromper a execução da aplicação.
Online Configuration Enabled	Selecione para permitir que as alterações feitas ao projeto sejam aplicadas imediatamente ao tempo de execução de **Test** ou **Startup**. Deve também estar conectado ao projeto em execução (ver definição de Status acima).

3. Clique em **Run Test** or **Run Startup** para iniciar a execução.

NOTA - As Ferramentas de Diagnose que podem ser marcadas para serem disparadas nas execuções do tipo Test e Startup, estão descritas em "Ferramentas para diagnose" na pagina 112

Construindo e Empacotando Projetos

A função de "build" é iniciada pressionando-se o ícone **Build** no painel sub-menu do ambiente de trabalho **Run.** Aparece a área de trabalho mostrada abaixo, para o controle da função.

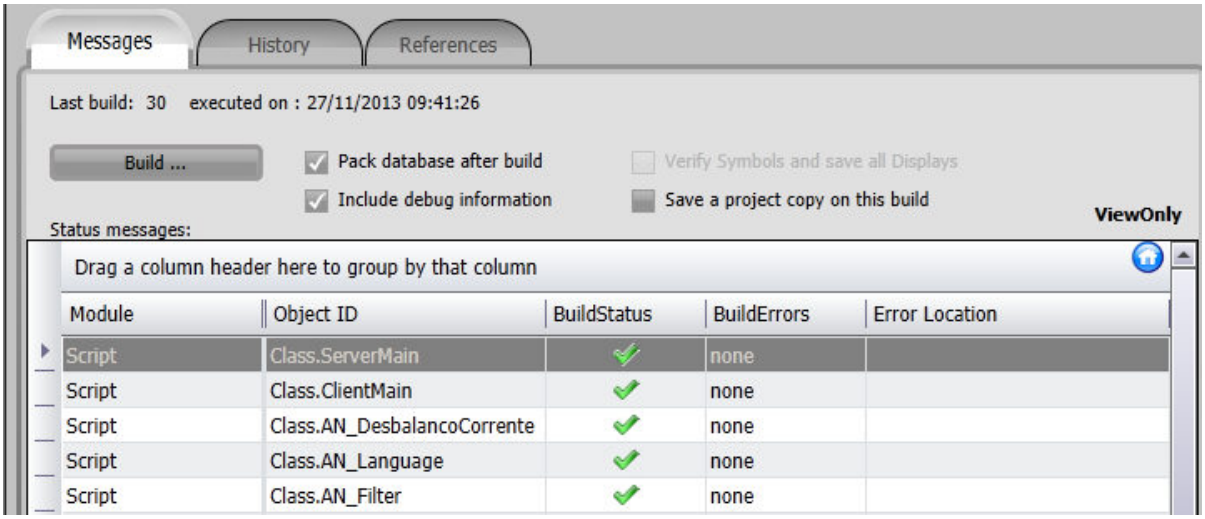
Build serve para recompilar todos os objetos telas e scripts do projeto para verificação final quanto a possíveis erros antes da colocação do projeto para a produção.

É feita uma verificação completa da lógica da aplicação durante a preparação dos arquivos para a versão final de produção. Esta operação não é necessária durante o desenvolvimento do projeto, já que todas as modificações são automaticamente e transparentemente sendo compiladas em "background" enquanto as edições são feitas.

A área de trabalho tem duas abas com tabelas informativas, não alteráveis. São os relatórios da atividade de Build.

Messages - Que mostra o status corrente de cada script e tela com informação sobre erros potenciais ou alertas sobre a lógica.Se você não executar um "Build" completo, a coluna BuildStatus, neste relatório reflete quaisquer avisos ou erros encontrados durante o processo de compilação em background.

Dê um duplo clique em uma linha com um X vermelho para ir para o código fonte onde está a linha que gerou este aviso ou erro. Os avisos são informativos e não parar o script seja executado. Erros evitar o script especificado de execução, mas não afetam a aplicação inteira. Se um script ou monitor tem um aviso, ele ainda será executado.



History - que lista registros de quando e quantas vezes foi executado um Build completo neste projeto.

References - É uma terceira aba que deve ser utilizada para o registro de Referências à arquivos de código externos ou do próprio usuário

---

Building projetos

Periodicamente, você deve executar uma compilação completa (Build):

- Quando você fez muitas mudanças e você quer uma validação completa e recompilação de todo o projeto.
- Quando você deseja atribuir um número de compilação para uma versão.
- Quando você quer "empacotar" o banco de dados. Quando o "build" é executado, o sistema cria um backup do arquivo de projeto atual. Se você deseja salvar o projeto como ele era antes desta compilação, renomeie o arquivo de backup.
- Quando você está se preparando para fazer uma publicação, ou seja, criar uma aplicação somente leitura para tempo de execução, em ambiente de produção.

Para criar o aplicativo de produção pelo Build:

1. Vá para **Run> Build> Messages**.
2. Se você desejar "empacotar" o banco de dados, selecione a opção "Pack database after build" (empacote o banco de dados após a execução do Build).

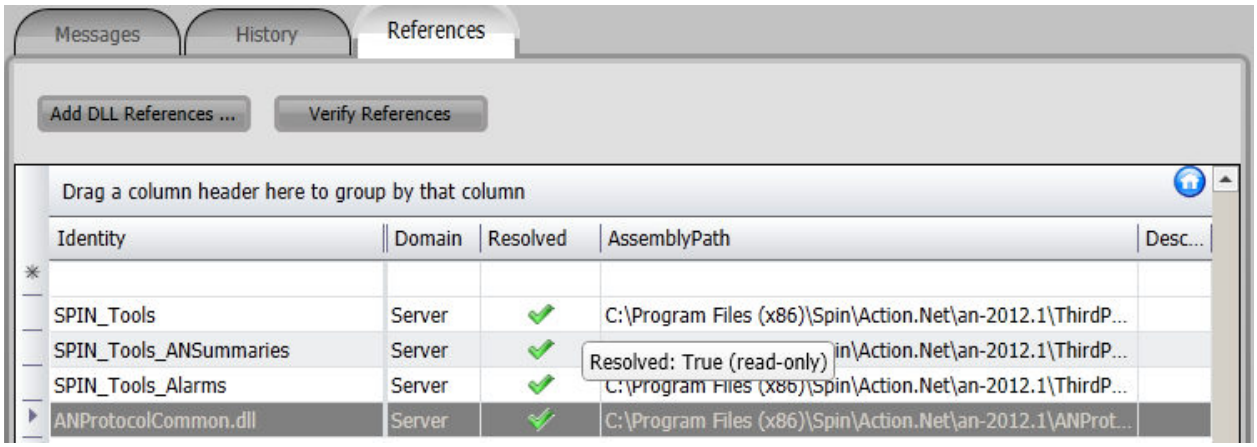
- O empacotamento, reduz significativamente o tamanho do arquivo de projeto. O sistema cria um arquivo com a extensão de backup, que é o banco de dados antes do empacotamento. Normalmente, você vai quer empacotar o banco de dados cada vez que você executar uma compilação.

3. Se você quiser salvar todas as telas, selecione a opção "Verify Symbols and Save all displays"
   - Certifique-se de usar esta opção se você tiver feito alterações para a biblioteca de símbolos. Esta opção se aplica a todas as modificações na biblioteca de símbolos ao longo do projeto.
   - A opção "include debug information" deverá ser selecionada se você desejar que após o Build ainda exista informação para a depuração de scripts.
4. Clique no botão **Build**.

Dica: Quando selecionada a opção "Save a project copy on this build", uma cópia de segurança do projeto envolvido nesta operação de **build** é criada automaticamente.

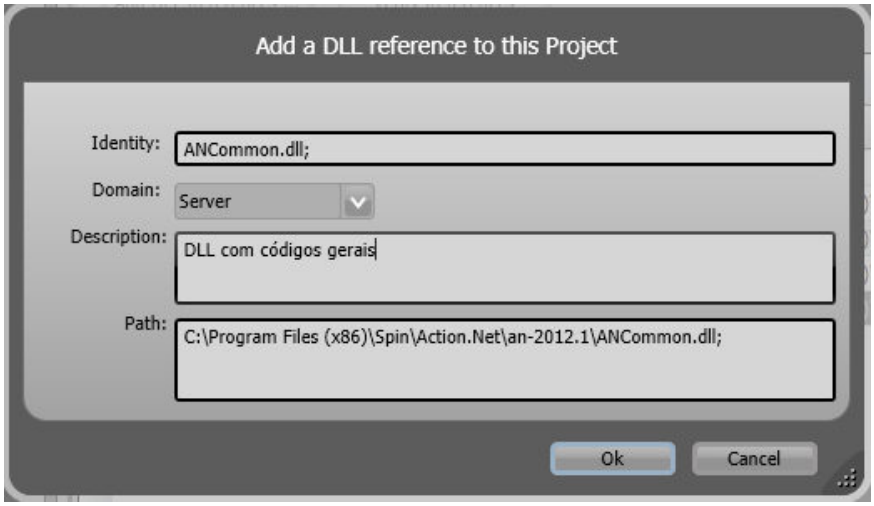
Referências à códigos externos

Na aba **References** do menu **Build** pode-se criar referências a DLLs externas ou verificar referências já existentes.



Para criar uma referência à uma DLL externa:

1. Vá para **Run>Build>References.**
2. Clique no botão **Add DLL reference.** Será mostrada a janela comum de MS-Windows para a seleção de um arquivo.
3. Navegue até onde está a DLL que voce quer fazer a referência e clique sobre ela.
4. Aparece então a janela de definição de Referência mostrada abaixo. Digite ou selecione as informações, conforme necessário.



Coluna	Descrição
Identity	Somente leitura- O nome da DLL escolhida na janela de procura de arquivos
Domain	Selecione o domínio para o uso da referência: • Server - DLL será utilizada por scripts no servidor • Client - DLL será utilizada por scripts nas máquinas clientes
Description	Digite uma descrição da referência
Path:	Somente leitura - Mostra o caminho completo onde se encontra a DLL à qual se fez esta referência.

Para verificar referências existentes:

1. Vá para **Run>Build>References**.
2. Clique no botão **Verify References.**
   - Serão mostradas no DataGrid todas as DLL externas já referidas nos scripts existentes no projeto.

Controle de Versão de Projetos

Nesta secção são apresentadas as funcionalidades existentes no Action.NET para o controle de versões do projeto.

Controle das alterações de configuração

O Action.NET implementa várias maneiras de ajudá-lo a manter o controle das mudanças de configuração do projeto:

- Todas as tabelas de configuração tem para cada registro, a DateCreated e as informações DateModified.
- O **Run-Build-History** registra e mostra todos os comandos BUILD executados para o projeto. Um backup do projeto para cada estado pode ficar disponível de acordo as configurações do usuário.
- O **Info> Track> Tables** mostra todas as tabelas de configuração, listando o número de linhas de cada uma e se foram alteradas desde a ultimo Build e o ultimo Publish.
- O **Info> Track> Changes** mostra todos os objetos do projeto que foram modificados. Para ativar ou desativar esta funcionalidade, você deve estar logado como administrador do aplicativo. Por default, o rastreamento de objetos só é ativado depois de uma operação de **Publish** (publicar) do projeto, mas você pode ativar a qualquer momento, independentemente de utilizar ou não o recurso de publicar.

Publicando o Projeto

Publish (publicar) o projeto cria uma versão somente leitura do projeto, que poderá ser colocada no ambiente final de produção sem o risco de puder ser alterada.

NÃO é necessário **publicar** o projeto para instalá-lo em um ambiente final de produção. Em cenários em que você pretende ainda ter mudanças contínuas no projeto já em campo, é mais simples colocar o arquivo do projeto principal, TPROJ, diretamente no computador de produção.

Os benefícios da publicação são os de o sistema criar uma versão compactada e somente leitura do arquivo de projeto. O arquivo criado tem o mesmo nome do arquivo de projeto, inclui um número da versão publicada e usa a extensão TRUN.

Os cenários típicos para a execução de comando **Publish** são as seguintes:

- Você quer implantar uma versão somente leitura do projeto, por exemplo, para estar em conformidade com ambientes certificados e regulamentados.
- Você quer usar o sistema de numeração de versão automática. O resultado da publicação é um arquivo TENG. Que também contém um número da versão, com um digito maior (1.0) e um menor(0.1) como parte do nome do arquivo. Além disso, **Info> Track** ajuda a gerenciar os arquivos publicados, incluindo o numero de Build atual do projeto.
- Você quer menor tamanho do arquivo do projeto e carregamento mais rápido do mesmo na partida, por exemplo, em máquinas, OEM e sistemas embarcados. O arquivo TRUN pode ser de até 5 a 10 vezes menor do que o arquivo do tipo TPROJ.
- Você quer proteger o projeto de modificações.

AVISO - O arquivo TRUN é sempre somente leitura, mas se a configuração do projeto não será visível para o usuário final, é uma opção independente definida no Sistema de Segurança. Se você não quer que os usuários finais vejam a configuração do projeto remova a permissão do usuário Guest e de outros usuários de Editar o projeto, antes de publicá-lo.

Para publicar o projeto:

1. Vá para **Run> Build> Messages**.
2. Selecione as configurações de compilação desejada e clique em **Build**.
3. Vá para **Run> Publish**.
   - Os campos de configurações do projeto atual mostram o status de somente leitura do projeto.
4. Selecione as Configurações de publicação que você deseja.
5. Clique **Publish**.

Objetos runtime Information

Há nomes de alguns objetos de tempo de execução que fornecem informações sobre a execução do projeto.

O namespace **Info** é o principal local para o estado do tempo de execução. Os principais objetos no namespace que são:

`Info.Project`: informações sobre o projeto que está sendo executado

`Info.License`: informações sobre a licença

`Info.Module`: Informação e comando start / stop para os Módulos

`Info.Trace ()`: exibe uma mensagem no TraceWindow

`Info.TestMode`: mostra se o projeto está sendo executado em modo de teste

O namespace **`Server`** tem informações sobre o computador servidor.

O namespace **`Client`** tem informações sobre cada computador cliente. executando uma interface gráfica.

Veja http://www.spinengenharia.com.br/help/an-2014/runtime/index.html , para a referência de programação completa em objetos de tempo de execução.

Implantando projetos Action.NET

Os projetos Action.NET podem ser implantados para executar localmente em um computador único ou em um dispositivo móvel, bem como utilizando uma arquitetura distribuída cliente-servidor ou até mesmo na nuvem.

As seções a seguir descrevem como implantar o aplicativo para a sua execução.

- "Sistemas locais e distribuídos" na pagina 203
- "Instalação do produto no computador de destino" na pagina 204
- "Licença e Verificação da configuração do Projeto" na pagina 204
- "Instalando arquivos de configuração do Projeto" na pagina 205
- "Configurando o servidor para Iniciar o Runtime" na pagina 206
- "Configuração para Clientes remotos" na pagina 210
- "Implantação de Sistemas Redundantes" na pagina 214

Sistemas locais e distribuídos

Quando você terminar de desenvolver e testar o projeto, você pode implantar o aplicativo em tempo de execução para ser utilizado pelos usuários finais.

O Action.NET pode ser implantado em várias arquiteturas diferentes, tais como:

- Incorporado à equipamentos sem interface de usuário local.
- Painéis ou computadores únicos (stand-alone) com interface de usuário local.
- Sistemas distribuídos com muitas estações clientes acessando um servidor.
- Servidores redundantes com vários usuários conectados.

Os procedimentos de configuração para cada arquitetura tem muitos passos comuns. Para todos os sistemas, até mesmo para as instalações autônomas, o Action.NET tem o conceito de componentes servidor e cliente.

Os componentes do Servidor são o Arquivo de Projeto e os módulos que executam as tarefas do lado do Servidor, como o de aquisição de dados,, tratamento de alarmes e registro de dados.

Os componentes Clientes são as Interfaces Gráficas de usuário e os scripts relacionados.

Quando você tem um projeto local ou em um único computador (stand-alone) os componentes do Servidor e do Cliente são executados na mesma máquina.

As tecnologias de cliente usadas pelo Action.NET simplificam muito a implantação já que você instala o projeto somente no computador servidor. Todas as estações clientes usarão o mesmo projeto deste servidor.

Para a instalação dos componentes do Servidor veja as seguintes seções:

- "Instalação do produto no computador de destino" na pagina 204.

- “Licença e Verificação da configuração do Projeto” na pagina 204.
- “Instalando arquivos de configuração do Projeto” na pagina 205.

Para a instalação dos componentes Clientes haverá alguma pequenas diferenças conforme o tipo de cliente a ser utilizado. O Action.NET suporta as seguintes tecnologias para os clientes:

- Windows Rich Client— Veja “Windows Rich Clients” na pagina 210
- Windows Smart Client—Veja “Windows Smart Clients” na pagina 211
- Windows Web-based—Veja “Windows Web Clients” na pagina 212
- iOS—Veja “Clientes iOS, iPhone e iPads” na pagina 213
- Active-X, COM, e JavaScript— você pode acessar a aplicação de tempo real usando a DataAccess API, que é uma interface COM para prover a integração com Active-X, JavaScript nas páginas web, ou linguagens de programação legadas como o VBScript.

Para arquiteturas redundantes veja “Implantação de Sistemas Redundantes” na pagina 214.

Instalação do produto no computador de destino

- Acesso a todos os PLCs referenciados e seus dados.
- Instalação completa do Action.NET, incluindo o servidor web embutido (TWebServer) ou IIS.

A instalação requer. o framework dot NET. Veja ““Obtendo Ajuda” na pagina 10.

O computador stand-alone ou os componentes do servidor em sistemas distribuídos exigirão que se que tenha o Action.NET instalado e licenciado.

A seção “Instalando o Action.NET” na pagina 12 descreve a maneira padrão de instalar e licenciar o Action.NET.

Quando você não estiver usando componentes que requeiram a configuração do Windows Registry, como os componentes OPC, também bastará copiar os arquivos do produto sem executar qualquer tipo de instalação (o que é muito útil para incorporar o software em dispositivos de dicados), remover exemplos de projetos ou componentes de engenharia, adicionar protocolos personalizados.

Em sistemas distribuídos, os computadores Cliente não requerem uma licença, eles só precisam ter a possibilidade de se conectar ao servidor. A limitação de uso é feita pelo número de usuários Clientes de tempo de execução, habilitado na licença do servidor. Para mais informações sobre a configuração do cliente, consulte “Configuração para Clientes remotos” na pagina 210.

Se o sistema implantado deve suportar usuários remotos, o TWebServer ou MS-IIS deverá ser instalado e estar em execução no computador servidor.

Licença e Verificação da configuração do Projeto

O projeto Action.NET é criado visando uma Família de Produtos e o Modelo de Produto específico, conforme definido na página **Info > Project > Settings** do aplicativo de Engenharia.

É necessário garantir que a licença no computador do servidor é **igual ou mais ampla** que aos requisitos do projeto, o que implica na seguinte lista de verificação:

- A família definida na licença deve ser a mesma que a família no projeto ou uma família de hierarquia superior.
  - Licenças **Enterprise** podem executar todos os projetos (Enterprise, HMI e OPCServer)
  - Licenças **HMI** podem executar projetos HMI e projetos OPCServer
  - Licenças **OPCServer** licenças podem executar apenas projetos OPCServer
  - Licenças ou projetos **Express** não são autorizados para ambientes de produção.
- O **Modelo** da licença deve suportar um número de pontos de comunicação igual ou superior ao numero de pontos do projeto que será executado no computador servidor.
- O **tipo de licença** no computador de destino deve ser **Engenharia** ou **Runtime**. As licenças do tipo DEV (Development) são apenas para Integradores de Sistemas e para trabalho interno e não devem ser usadas em ambientes de produção.
- Se o projeto requer quaisquer interfaces adicionais, como OSIsoft(tm) PI System, ou protocolos padronizados IEC ou DNP, certifique-se que a licença no computador de destino está habilitada para essas interfaces.
- Verifique se o numero de clientes remotos habilitado na licença do Servidor comporta os seus requisitos de projeto.

Para mais informações sobre licenças Famílias de Produtos e Modelos de Produto, consulte "Versões e Licenças Action.NET" na pagina 217.

Instalando arquivos de configuração do Projeto

A configuração do projeto está completa em um único arquivo, com extensão TPROJ ou TRUN, conforme você deseja instalar para a atividade em produção do arquivo principal de configuração do projeto ou uma versão publicada, com garantia de não poder ser alterada.

O utilitário de Gerenciamento de Projetos permite você se conectar com servidores remotos e baixar os arquivos de projeto para estes computadores remotos.

Apesar de um único arquivo conter a configuração do projeto inteiro, você deve usar a seguinte lista de verificação para garantir que todas as dependências externas também estão atendidas.

- Se as estruturas de pastas no computador de Produção não são as mesmas que você usou para o desenvolvimento, certifique-se de que todas as referências que você tem à caminhos de arquivos em seu projeto estão mapeadas corretamente para o computador de produção. O Action.NET tem muitos recursos e macros para definir os caminhos relativos à localização do projeto, instalação do produto, ou caminho dos aplicativos execução; Sempre que possível, utilize estas macros e evite usar definições fixas de caminho de arquivos seus projetos.
- Os controles externos WPF também devem ser copiados no computador de destino. Para o acesso remoto web esses arquivos devem estar localizados na pasta WpfControl e deve ser executado o utilitário para atualizar o "web manifest".

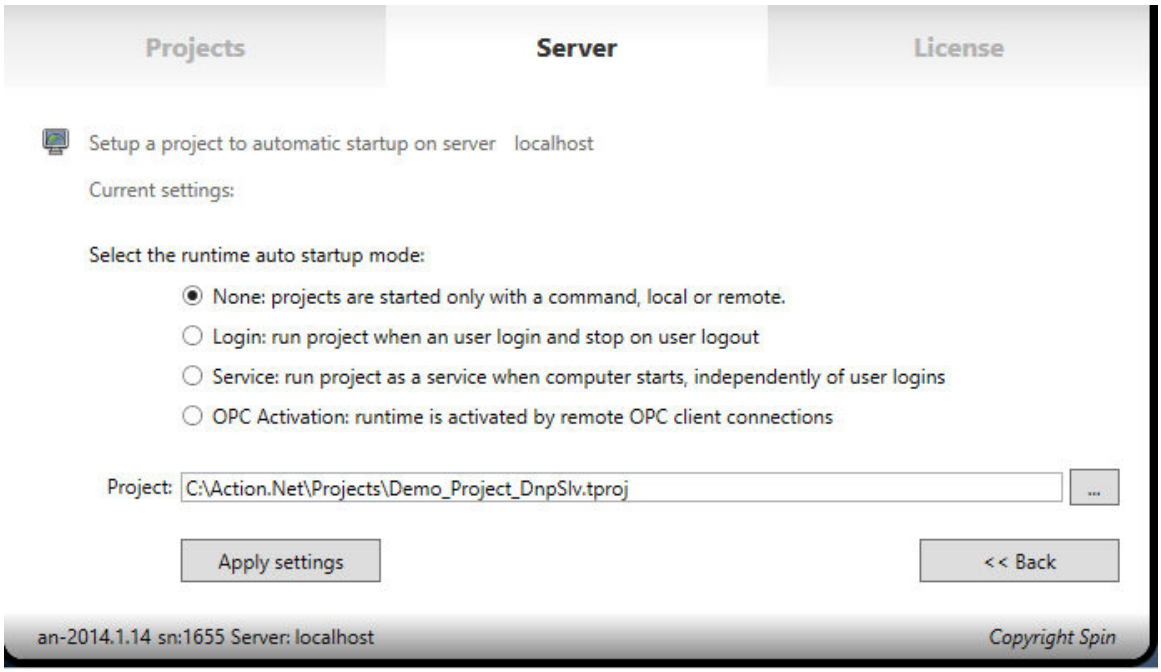
- Se o aplicativo utiliza referências externas à DLLs. ou "assemblies" dot NET, garanta que estes arquivos estão disponíveis e os caminhos estão corretos no computador de destino.
- Se o projeto utiliza valores retentivos, você deve decidir se será criado um novo banco de dados Retentivo no computador de destino ou se você vai copiar um banco de dados com alguns valores já pré-definidos.
- Deve-se habilitar o Firewall para permitir o acesso de clientes remotos: As portas 3101 para execução tipo **Startup** ou opcionalmente a porta 3201 para o modo de **Test**. Para clientes web e iOS, os serviços de dados da web devem estar habilitados na porta 80.
- Se o aplicativo estiver usando DLLs externas, WpfControls, arquivos de configuração ou bancos de dados incorporados, certifique-se de copiar os arquivos para o computador de destino e verifique se os utilitários do Gerenciador de Projetos permitem conectar com servidores remotos e baixe os arquivos do projeto para os computadores remotos.

Configurando o servidor para Iniciar o Runtime

Você pode executar o projeto em qualquer uma das seguintes formas:

- Iniciar manualmente, a partir da janela principal do Action.NET e da lista de projetos: clique como botão direito no projeto e selecione no menu **Run Project** .
- Iniciar manualmente, quando configurando o projeto, no aplicativo de Engenharia: vá para **Run > Startup** e clique em **Run Startup**.
- Iniciar o projeto automaticamente, o que é a melhor opção para ambientes de produção.

A melhor maneira de definir uma inicialização automática do produto, é usar a interface de configuração disponível nas ferramentas do Gerenciamento de Projetos, na aba **Server**.



Recomenda-se iniciar manualmente durante o desenvolvimento do projeto ou instalação, neste caso é necessário um operador durante o processo de inicialização. Considera-se que a inicialização automática é o modo mais adequado para computadores já em atividade de produção.

Você deve utilizar a opção de **Login**, e o atalho de inicialização, ao testar o projeto ou em arquiteturas onde a aplicação irá executar sempre no mesmo perfil de usuário do Windows.

Use o **Service** do Windows em servidores de produção quando você precisar de diferentes usuários do Windows para iniciar sessão no computador, mantendo os componentes de tempo de execução do servidor indefinidamente em execução.

O Action.NET suporta a ativação do aplicativo a partir de um cliente OPC, para estar em completa conformidade com as especificações do OPC. Não recomendamos seu uso, pois, como um princípio de boa prática, não é desejável permitir que o cliente remoto possa controlar se o aplicativo do lado do servidor está sendo executado ou não.

As seções a seguir descrevem o que essas configurações automáticas estão de fato fazendo. Se, por algum motivo, você precisar configurar a inicialização sem usar as ferramentas de configuração estes comandos são os que devem ser utilizados.

Usando um atalho na inicialização do Windows

Você pode configurar um servidor Windows para iniciar automaticamente um projeto usando um atalho de inicialização. O atalho de inicialização só inicia a aplicação quando um usuário faz login no Windows e a aplicação pára de funcionar quando o usuário faz logoff do Windows.

Este procedimento é executado automaticamente pelo sistema quando o modo de inicialização for selecionado, conforme descrito em "Configurando o servidor para Iniciar o Runtime" na pagina 206. Esta seção irá explicar como configurar manualmente o os atalhos.

O início de execução do projeto é realizado pelo programa **TStartup.exe**.

NOTA - Ao usar este exemplo, certifique-se de alterar o caminho de instalação e a versão do Action.NET para os que de fato foram instalados em seu computador.

Parâmetros de linhas de comando:

/project: Projeto Caminho e nome entre aspas duplas

/ username: (opcional), nome do usuário que será usado para iniciar o servidor, se você não especificar, o usuário GUEST será usado.

/redundancy: indica que a redundância do servidor está sendo usado (requer IP1 e IP2)

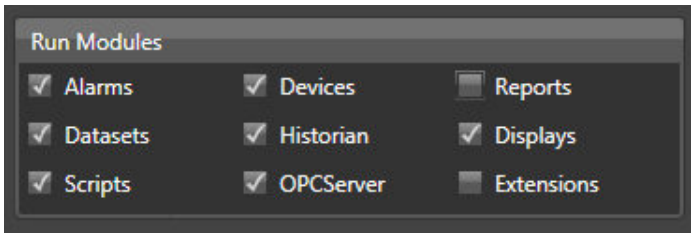
/ ip1: Endereço IP do servidor principal Action.NET

/ Port1: porta TCP do servidor principal Action.NET

/ IP2: Endereço IP do servidor secundário Action.NET

/ Port2: porta TCP do servidor secundário Action.NET

Os módulos que serão iniciados são os configurados no projeto em **Run.Startup**.



Exemplos:

"C:\Program Files (x86)\Spin\Action.Net\an-2014.1\TStartup.exe" /project:"C:\ActionNet Projects\Project1.tproj"

"C:\Program Files (x86)\Spin\Action.Net\an-2014.1\TStartup.exe" /project:"C:\ActionNet Projects\Project1.tproj" /port1:3101

Redundancia de Servidores:

"C:\Program Files (x86)\Spin\Action.Net\an-2014.1\TStartup.exe" /project:"C:\ActionNet Projects\Project1.tproj" /username:Administrator /redundancy /ip1:192.168.1.1 /port1:3101 / ip2:192.168.1.2 /port2:3101

Criando o atalho

Vá para **Start > All Programs,** e clique com o botão direito na pasta de inicialização, e selecione **Open.**

1. Na pasta **Inicializar**, clique com o botão direito e selecione **Novo> Atalho**.
2. Na janela Criar Atalho, cole no campo o comando acima.
3. Se você não estiver usando redundância, exclua a parte redundância do texto.

> **AVISO -** Nos exemplos abaixo, não se esqueça de alterar o caminho de instalação e a versão do Action.NET para as da instalação real em seu computador.

4. Digite ou cole a linha completa de comando, exemplos: "C:\Program Files (x86)\Spin\Action.Net\an-2014.1\TStartup.exe" /project:"C:\ActionNet Projects\Project1.tproj"
5. Clique em Avançar.
6. Digite um nome para o atalho.
7. Clique em Concluir.
   - Quando você reiniciar o computador, o projeto será iniciado automaticamente.

Usando um Serviço do Windows para inicialização

Você pode configurar um servidor Windows para iniciar automaticamente um projeto quando o computador for inicializado, utilizando um **Serviço** do Windows. O serviço Windows inicia o aplicativo assim que o computador é ligado e o sistema operacional Windows é iniciado, mesmo se nenhum usuário tiver feito logon no Windows.

Este procedimento é executado automaticamente pelo sistema quando selecionado para o modo **Satrtup**, como descrito "Configurando o servidor para Iniciar o Runtime" na pagina 206. Esta seção explicará como configurar-lo manualmente.

NOTA - Estes métodos não iniciam o cliente (interface do usuário com os monitores). Para automatizar a inicialização do cliente, ""Iniciando automaticamente clientes Windows" na pagina 212

Use o Serviço do Windows somente em servidores de produção que você não está usando como estações de engenharia, e apenas se você precisar a capacidade de diferenciar os usuários do Windows ao fazer o login, enquanto o projeto está sendo executado.

As primeiras etapas do procedimento a seguir são necessárias para você configurar o Serviço do Windows.

Para executar o aplicativo como um serviço do Windows:

1. Vá para **Info> Project> Redundancy**.
2. Digite ou selecione as informações, conforme necessário.

Campo	Descrição
Enable Configuration	Selecione para habilitar esta configuração
Primary Server IP and Port	Entre com o endereço IP e a porta do servidor primário
Secondary Server IP and Port	Entre com o endereço IP e a porta do servidor secundário, se houver.
On Primary Startup	Selecione a opção desejada.
Historian Replication	Selecione como deve ser tartada a a replicação do historiador.
Connection Timeout	Tempo máximo permitido para conexão, em segundos, antes de chavear para o servidor secundário.
Server Command Line	Campo de somente leitura carregado de acordo com as opões dos campos anteriores. Clique em C**opy to Clipboard** para copiar o comando para ser usado em atalhos.

3. Ao lado do campo de linha de comando do servidor, clique em **Copy to Clipboard**.
   - Vai copiar o campo somente leitura, apresentado com base nos campos acima dele.
4. Abra um arquivo de texto, cole, e continue com os passos 5 até 7 neste arquivo.

5. Se você não estiver usando redundância, deve excluir a parte sobre redundância do texto.
6. No início da linha de comando, digite ou cole o caminho completo para a pasta de instalação do Action.NET e coloque aspas em torno do texto
    - Deverá ficar algo como o mostrado a seguir:

```
"C:\Program Files (x86)\Spin\Action.NET\an-2014.1.3\tStartup.exe"
```

7. Antes do texto "tproj", digite ou cole o caminho completo para o projeto.
    - Deve ficar parecido com isto:

```
/project:C:\Action.NET Projects\<project_name>.tpro
```

    - A linha de comando completa deve ser algo como isto:

```
"C:\Program Files (x86)\Spin\Action.NET\an-2014.1.3\tStartup.exe" /
project:/project:C:\Action.NET Projects\<project_name>.tproj /
username:<username>
```

8. Deixe o arquivo de texto aberto para uso no passo 10.
9. No prompt do DOS, vá para o <Caminho de Instalação. do NET Framework > e execute o seguinte comando:

```
installutil <InstallPath>\<an-version>\TStartupAsService.exe
```

    - Exemplo:

```
C:\Windows\Microsoft.NET\Framework\v2.0.50727>installutil c:\Program
files\Spin\Action.NET\an-2014.1\TStartupAsService.exe
```

10. Copie e cole o comando que você criou no arquivo de texto.
11. . No Windows Registry, configure os parâmetros em:

    "HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\TStartup\Image Path"

    - Example:

```
"c:\\Spin\an-2014.1\TStartupAsService.exe" "/project:C:\Action.NET
Projects\Project1.tproj""HKEY_LOCAL_MACHINE \ SYSTEM \ CurrentControlSet
\ services \ TStartup \ ImagePath"
```

12. No Windows Services (em Ferramentas Administrativas), defina "TStartup Service" como Automático, de modo que o projeto selecionado será iniciado quando o computador é iniciado.

Configuração para Clientes remotos

Windows Rich Clients

Você pode implantar seu aplicativo para uso por Windows Rich Clients. A Tabela 1 descreve como a implementação por Windows Rich Client funciona.

Tabela 1: Instalação do Action.NET para usar Windows rich client

Instalação	Instalar o Action.NET no computador do cliente. Veja "Instalando o Action.NET" na pagina 12.
Como iniciar	Execute o programa TRichClient.exe. Para um exemplo de como para iniciar automaticamente o cliente quando um usuário efetua login no Windows, consulte "Iniciando automaticamente clientes Windows" na pagina 212.
Execução	É executado em sua própria janela. Isto permite uma forte segurança de usuário, incluindo a capacidade de desativar o mecanismo de chaveamento de tarefas do Windows, de acordo com o login do projeto em execução. Ao executar o TRichClientt.exe em máquinas de 64 bits, o aplicativo é executado em código nativo de 64 bits. Se você precisar executar a versão de 32 bits, por exemplo, para garantir a compatibilidade com componentes legados COM e Active-X COM, você pode usar o programa TRichClient32.exe.
Comunicação	Comunica-se com o servidor usando WCF (porta configurável, default 3101).

Windows Smart Clients

Você pode implantar seu aplicativo para uso na forma de Windows Smart Clients (cliente inteligente).

O Smart Client funciona como o Rich Client, ou seja, eles funcionam da mesma maneira, mas o cliente inteligente utiliza a instalação ClickOnce ™. Esta tecnologia permite que você tenha a mesma funcionalidade que o Rich Client, mas sem ter que instalar o Action.NET em seu computador.

A primeira vez que você acessar o aplicativo, o sistema transfere automaticamente os componentes necessários para executar o aplicativo. A próxima vez que você acessar o aplicativo, o sistema verifica se o cache local é a mesma versão do aplicativo que está no servidor e, se necessário, atualiza o cache local antes de executar o aplicativo. Se a versão é a mesma, o aplicativo é iniciado imediatamente.

A Tabela 2 descreve como funciona a implantação e execução de Smart Client.

Tabela 2: Implantação do Action.NET como Windows Smart Client

Instalação	Não requer instalação. Você só precisa dotNET Framework 4.0 e Internet Explorer 8.0 ou posterior no computador do cliente. A primeira vez que você iniciar o aplicativo, ele irá baixar automaticamente os componentes necessários a partir do servidor. Toda vez que o aplicativo é iniciado, ele verifica automaticamente se uma nova versão está disponível no servidor.
Como iniciar	A partir do Internet Explorer (ou um atalho) ir para o URL: http://<ServerIPAddressorName>/an-2014.1/TSmartClient.application Para um exemplo de como iniciar automaticamente o cliente quando um usuário faz login no Windows, consulte "Iniciando automaticamente clientes Windows" na pagina 212
Execução	Funciona exatamente da mesma forma que o Rich Client. As funcionalidades do Rich Client e o SmartClient são as mesmas, apenas a instalação e os métodos de ativação são diferentes.
Comunicação	Comunica-se com o servidor usando WCF (porta configurável, default 3101).

Windows Web Clients

Você pode implantar seu aplicativo para uso como clientes Web do Windows.

A Tabela 3 descreve como a implementação do Web Client funciona.

Tabela 3: Implantação do Action.NET como Windows web client

Instalação	Não requer instalação. Você só precisa o dot NET Framework 4.0 e Internet Explorer 8.0 ou posterior no computador do cliente. A primeira vez que você iniciar o aplicativo, irá fazer o download automaticamente a partir do servidor dos componentes necessários. Toda vez que o aplicativo é iniciado, ele verifica automaticamente se uma nova versão está disponível no servidor.
Como iniciar	A partir do Internet Explorer (ou um atalho) ir para o URL: http://<ServerIPAddressorName>/an-2014.1/TWebClient.Xbap Para um exemplo de como iniciar automaticamente o cliente quando um usuário faz login no Windows, consulte "Iniciando automaticamente clientes Windows" na pagina 212
Execução	Roda dentro de uma janela do navegador web usando "Confiança Parcial" (Sandbox Security).
Comunication	Comunica-se com o servidor usando HTTP ou HTTPS (porta 80).

Iniciando automaticamente clientes Windows

Você pode criar um atalho para o executável apropriado, dependendo do tipo de cliente, para iniciar automaticamente o aplicativo em um cliente Windows. Você pode criar o atalho na área de trabalho ou colocá-lo na pasta de inicialização, conforme descrito abaixo.

Para o cliente web, você também pode configurar o aplicativo como a página inicial no Internet Explorer.

Vá para Iniciar> Todos os Programas

1. Clique com o botão direito na pasta de inicialização do Windows.
2. Selecione Abrir para abrir a pasta de inicialização.
3. Na pasta de inicialização, clique com o botão direito e selecione **Novo> Atalho**.
4. Na janela **Criar Atalho**, cole no campo que exibe.
5. Se você não estiver usando redundância, excluir a parte de redundância do texto.
6. Proceda de uma das seguintes formas:
   - Para um Rich Client - No início da linha de comando, digite ou cole o caminho completo para a pasta de instalação Action.NET e coloque aspas em torno dele.
     - Deve ser algo assim:

"C:\Program Files (x86)\Spin\Action.NET\an-2014.1.3\TRichClient.exe" /ip1:<IP_address>

- Para um Smart ou Web client - No início da linha de comando, digite ou cole o caminho completo para a pasta de instalação do Internet Explorer e coloque aspas em torno dele.
  - Deve ser algo assim:

"C:\Program Files (x86)\Internet Explorer\iexplorer.exe" http:<IP_address>/an-2014.1/TSmartClient.application

"C:\Program Files (x86)\Internet Explorer\iexplorer.exe" http:<IP_address>/an-2014.1/TWebClient.Xbap

7. Clique em **Avançar.**
8. Digite um nome para o atalho.
9. Clique em **Concluir**.
   - Na próxima vez que reiniciar o computador, o projeto será iniciado automaticamente.

Clientes iOS, iPhone e iPads

Você pode implantar seu aplicativo para uso por clientes iOS: iPhone, iPad, e iTouch. Para outros tablets, entre em contato com o suporte.

A Tabela 4 descreve como funciona a instalação em iOS.

Tabela 4: Instalação do Action.NET como cliente iOS

Instalação	Instale o aplicativo cliente SCADA HMI da Apple Store.
Como iniciar	Inicie o aplicativo cliente SCADA HMI e siga as opções de configuração inicial.
Execução	No iOS, o aplicativo roda nativamente, proporcionando assim maior desempenho, maior segurança e acesso a componentes gráficos nativos, em comparação com outros aplicativos usando Terminal Cliente, Remote Desktop, ou HTML web.
Comunicação	Comunica-se com o servidor, chamando um serviço da web usando a porta 80. O servidor deve estar na mesma rede local ou VPN que o dispositivo iOS, ou pode ter um endereço IP público, desde que o acesso HTTP esteja ativado.

Para implantar em um dispositivo iOS:

1. Desde o seu dispositivo iOS, toque no ícone da App Store.
   - Você também pode ir para a Apple App Store do iTunes.
2. Procurar e instalar o aplicativo cliente SCADA HMI.
3. Inicie a aplicação Cliente SCADA HMI.
4. Insira as seguintes informações:

Campo	Descrição
Host Server	Endereço IP ou nome do servidor de projeto.
Port	Porta 80.
Polling	Define a taxa de atualização entre o cliente e o servidor, expressa em quartos de um segundo. O valor padrão de 1, significa que o cliente recebe novos dados do servidor a cada 250 ms. Ao se conectar a servidores localizados nas redes de Internet ou de baixa largura de banda, esse valor deve ser aumentado. Para mais informações, consulte a ajuda do aplicativo.
User	Nome de Usuário como configurado no projeto. O padrão é GUEST
Password	Senha associada ao nome de usuário no projeto..
Project	Nome do projeto no servidor remoto.

5. Toque em **Login**.
   - É feito o "download" dos gráficos e telas, e então o aplicativo é iniciado. O aplicativo ao iniciar pela primeira vez pode demorar um pouco mais do que nas inicializações subsequentes.

Implantação de Sistemas Redundantes

Configuração das opções de redundância

Você pode configurar a redundância da aplicação configurando dois computadores como servidores. Um computador será o primário, e o outro será o secundário ou "Hot Stand By". Se o computador primário ou a sua conexão com o computador falhar, o sistema chaveia automaticamente para o computador secundário.

NOTA - Se você selecionou HMI como a Família de Produtos, a configuração de redundância não estará disponível.

Para configurar a redundância:

1. Vá para **Info> Project> Redundancy**.
2. Digite ou selecione as informações, conforme necessário.

Campo	Descrição
Enable Configuration	Selecione para ativar a configuração de redundância.
Primary Server IP and Port	Digite o endereço IP e a porta do servidor primário.
Secondary Server IP and Port	Digite o endereço IP e a porta do servidor secundário.
On Primary Startup	Selecione a opção desejada.
Historian Replication	Escolha como tratar com a replicação do historiador.
Connection Timeout	Tempo limite de falha de conexão, em segundos, para mudar para o servidor secundário.
Server Command Line	Campo somente leitura preenchido com base nos campos acima. Clique C**opy to Clipboard** para copiar o comando para o uso.
Rich Client Command	Campo somente leitura preenchido com base nos campos acima. Clique C**opy to Clipboard** para copiar o comando para o uso.
Smart Client URL	Campo somente leitura preenchido com base nos campos acima. Clique C**opy to Clipboard** para copiar o comando para o uso.
Web Client URL	Campo somente leitura preenchido com base nos campos acima. Clique C**opy to Clipboard** para copiar o comando para o uso.

Pra iniciar automaticamente o aplicativo em um cliente Windows:

1. Vá para **Info> Project> Redundancy**..
2. Digite ou selecione as informações, conforme necessário.

Field	Descrição
Enable Configuration	Selecione para ativar a configuração de redundância.
Primary Server IP and Port	Digite o endereço IP e a porta do servidor primário.
Secondary Server IP and Port	Digite o endereço IP e a porta do servidor secundário.
On Primary Startup	Selecione a opção desejada.
Historian Replication	Escolha como tratar com a replicação do historiador.
Connection Timeout	Tempo limite de falha de conexão, em segundos, para mudar para o servidor secundário.
Rich Client Command	Campo somente leitura preenchido com base nos campos acima. Clique C**opy to Clipboard** para copiar o comando para o uso.

Field	Descrição
Smart Client URL	Campo somente leitura preenchido com base nos campos acima. Clique C**opy to Clipboard** para copiar o comando para o uso.
Web Client URL	Campo somente leitura preenchido com base nos campos acima. Clique C**opy to Clipboard** para copiar o comando para o uso.

3. Clique no botão **Copy** ao lado do tipo de cliente que você deseja iniciar automaticamente

Versões e Licenças Action.NET

O Action.NET tem características flexíveis que permitem gerenciar a versão do produto e o seu licenciamento

- "Números de versão do Action.NET" na pagina 217
- "Tipos de Licenças Action.NET" na pagina 218
- "Família e Modelo do Produto" na pagina 218
- "Acesso remoto à Projetos" na pagina 219
- "Configuração do projeto e Licenças" na pagina 220
- "Usando várias versões Simultaneamente" na pagina 220

Números de versão do Action.NET

A versão Action.NET é uma sequência de letras e números como: **an-2014.1.10**.

O significado dessa sequência é a seguinte:

<Ident_Produto> - <Versão principal> <Versão secundária> <Atualização>.

Portanto, neste exemplo:

- Identificação do produto = an
- Versão principal = 2014
  - A versão principal é um grande lançamento, que acontece a cada um ou dois anos. Ela inclui as principais alterações na documentação ou características do produto. Os clientes existentes devem ter o plano de manutenção ou comprar uma atualização para ter acesso a ela.
  - A nova versão instala para uma nova pasta, por isso não afeta qualquer projeto ou instalação do produto das versões anteriores. A proteção de licença, ou hardkey ou tecla de função (dongle), deve ser atualizada para usar essa versão.
- Versão secundária = 1
  - A versão secundária geralmente é lançado a cada três a nove meses e não é verificada pela licença. Ela inclui melhorias e correções do produtos.
  - Uma versão secundária é semelhante ao que às vezes é chamado de "Service Pack". No entanto, com o Action.NET, você pode ter várias versões secundárias instaladas em pastas diferentes e usar todas elas ao mesmo tempo no mesmo computador.
- Atualização = 10
  - O número de atualização reflete pequenas modificações em cima da versão atual. As atualizações são sempre compatíveis com a versão atual.
  - Atualizações são instalados sobre a versão do produto que se dirigem. Se você tentar abrir um projeto criado com uma atualização que você não tem no seu computador,

você pode executar esse projeto, mas a ferramentas de Engenharia será aberta no modo "read-only". A menos que você tenha uma necessidade ou problema específico que você necessita da atualização, você não é obrigado a fazer esta atualização.

Tipos de Licenças Action.NET

Você pode verificar a sua licença na aba **Licença** da página principal ou ao editar qualquer projeto em **Info > License > Current.**

Para obter informações sobre como instalar ou atualizar uma licença, consulte "Licenciando o Action.NET" na pagina 14.

Família e Modelo do Produto

Action.NET em diferentes famílias de produtos e modelos que atendem os requisitos para diferentes mercados. Sua licença define o grau mias alto de Família de produtos e Modelo que você pode definir para um projeto. Você sempre pode criar projetos que utilizam uma família ou modelo "menor".

- Action.NET Framework (Action.NET para uso com o PI System)
- Action.NET Enterprise
- Action.NET HMI
- Action.NET OPC-Server
- Action.NET Express

Se você não tem uma licença, o sistema automaticamente define os projetos criados como sendo da Família Express, que é apenas para avaliação e treinamento. Durante a execução o aplicativo "Runtime" para automaticamente depois de um cero período de tempo. Usando a versão Express, você pode apenas editar e executar projetos criados com a família de produtos Express. Ao editar um projeto, uma mensagem é exibida no canto superior direito quando você está executando a versão Express.

Veja o site do produto que tem a informação atualizada sobre a descrição e as diferenças entre as famílias de produtos: http://www.spinengenharia.com.br

O Modelo do produto definido na sua licença define o número máximo de pontos de comunicação e tags permitidos. Para mais informações, consulte "Executando Projetos" na pagina 54.

Family	Dependendo da sua licença Action.NET, você pode ter opções aqui para construir aplicações para uma versão específica do Action.NET. Tenha certeza que você sabe qual a versão runtime que seus usuários finais precisarão usar. A opção que você selecionar determina quantos pontos de comunicação você pode criar para o projeto: • Enterprise - Usar para aplicações maiores e mais complexas, com numero ilimitado de pontos de comunicação em tempo real. • HMI- Usar para aplicações menores que gerenciam hardware com até 1.500 pontos de comunicação em tempo real. • Express - Usar somente para fins de teste ou demonstração. • OPCServer - Usar quando você sabe que vai utilizar apenas comunicação por funções OPC. Você pode alterar essa configuração mais tarde, se necessário.
Model	Dependendo da sua licença Action.NET, você pode ter opções aqui para construir aplicações para um modelo de produto específico. Escolha qual o modelo com base no número de pontos de dados em tempo real, que você vai ter nos projetos. • Unlimited-Não há limite para o número de pontos de comunicação em tempo real. Sua memória de computador e desempenho são as únicas limitações. • ServerLarge-Até 100.000 pontos de comunicação em tempo real. • ServerMedium-Até 50.000 pontos de comunicação em tempo real. • ServerSmall-Até 25.000 pontos de comunicação em tempo real. • Plant-Até 15.000 pontos de comunicação em tempo real. • Supervisor-Até 5.000 pontos de comunicação em tempo real. • Line-Up para 2.500 pontos de comunicação em tempo real. • Work Cell-Até 1.500 pontos de comunicação em tempo real. • Máchine-Up para 500 pontos de comunicação em tempo real. • Panel - até 300 pontos de comunicação em tempo real. • Instrument - até 150 pontos de comunicação em tempo real. • Express - até 75 pontos de comunicação em tempo real. Você pode alterar essa configuração mais tarde, se necessário.

Acesso remoto à Projetos

Além da família de produtos e modelo de produto, sua licença também tem um tipo de licença:

- Engenharia - Permite configurar e executar projetos.
- Runtime - Permite executar projetos.
- Developer - Disponível apenas para parceiros VAR para criar e executar projetos em cenários de teste, e não para a produção.

Quando você se conecta a um computador remoto para abrir ou executar um projeto, a licença que se verifica é a licença no servidor, não a licença em seu computador local.

No entanto, se você se conectar a um computador remoto que tem licença de tipo Runtime somente, mas você tem uma licença de engenharia em seu computador, você será capaz de configurar e executar o projeto remoto (desde que você tenha também as senhas de segurança do projeto e autorização). Isto é útil para permitir que os engenheiros possam tratar com alterações necessárias no campo.

Configuração do projeto e Licenças

Ao editar um projeto, as informações sobre a família de produtos mostradas em **Info > License >Current** refletem a licença no seu computador (ou no servidor se você está abrindo o projeto a partir de um servidor).

A família do projeto e modelo de projeto em **Info> Project> Version** refletem as definições para esse projeto específico quee podem ser alteradas a partir de **Info> Project> Settings** .

Os recursos e funções disponíveis em um projeto são baseados na família do projeto especificada para o projeto (conforme definido em **Info > Project > Version**), e não na sua licença. Se você tiver uma licença Enterprise, mas um projeto foi criado com a família de projeto definido para HMI, o sistema só permitirá que você use os recursos e funções que são válidas para a família de produtos HMI.

Se você tentar abrir um projeto que tem uma exigência maior do que a licença que você tem, você não será permitido e uma mensagem no lado superior direito será exibida.

Usando várias versões Simultaneamente

As versões do Action.NET com as numerações de Versão principal ou Secundária diferentes são instaladas em pastas diferentes das versões anteriores. Isso permite manter diferentes versões do Action.NET simultaneamente instaladas no mesmo computador.

A estrutura da pasta é:

- \Action.NET— Pasta com a seleção de projetos e ferramentas de gestão
- \Action.NET\an-2014.1— Arquivos específico para a versão an-2014.1.
- \Action.NET\an-2012.1— Arquivos específico para a versão an-2012.1.

Você vai ter tantas sub-pastas quantas forem as versões do produto Action.NET que estiverem instaladas em seu computador.

NOTA - Devido o espaço da instalação ser muito pequeno, apenas cerca de 100 MB, recomendamos que você mantenha todas as versões. Ao abrir um projeto, o sistema verifica automaticamente qual a versão do projeto em que foi criado, e ele vai procurar e abrir a com as

ferramentas que correspondem ao projeto.

Se você tentar abrir um projeto criado com uma nova versão do produto (por exemplo, você tem an-2012.1 e está tentando abrir um projeto criado com a versão 2014.1), o sistema não irá abrir ou executar aquele projeto. No canto superior direito do seu espaço de trabalho, você verá uma mensagem explicando por que o projeto não pode ser aberto.

Se você tentar abrir um projeto usado pela última vez com uma atualização mais recente, por exemplo, você tem a versão an-2014.1.10 e tentar abrir um projeto que foi usado com versão an-2014.1.20, desde que você tenha a licença para an-2014, você ainda é capaz de executar esse projeto, e abrir o projeto com a ferramenta de Engenharia, mesmo que o projeto está usando uma atualização que é mais recente que o que você tem em seu computador. Para proteção e para evitar inconsistências, não será permitido que o projeto seja modificado, porém, neste cenário que se abre em modo somente leitura. No lado esquerdo superior direito da sua ferramenta de engenharia você tem uma mensagem indicando este modo quando ocorre esse cenário.

Se você tem um projeto que foi criado em uma versão anterior do produto (seja versão principal ou secundária), você pode atualizá-lo para a versão que você está usando atualmente. Para mais informações, consulte "Atualizando projetos" na pagina 54

Namespaces de tempo de execução

Objetos runtime Project

Todos os componentes do projeto, tais como tags, alarmes, dispositivos têm associados classes e objetos dot NET com propriedades e métodos acessíveis pelo sistema de tempo de execução, como explicado em "Modelo de Objetos e Namespaces" na pagina 29. A documentação. NET para os espaços de nomes, seguindo o layout do MSDN, é acessível em http://www.spinengenharia.com.br/help/an-2014/runtime/index.html.

Classes do Microsoft .NET Framework

O Action.NET tem acesso a todos os recursos do Microsoft dotNET Framework, como uma aplicação de código gerenciado puro. Para obter informações sobre cas classes C # e VB.NET refera-se a Microsoft MSDN. Framework 4 documentação on-line Net em http://msdn.microsoft.com/en-us/library/gg145045(v=vs.100).aspx

Script Toolkit

Ao criar código de scripts dentro do Action.NET, pode-se utilizar um conjunto de métodos utilitários públicos, a que chamamos Script Toolkit. Esses métodos estão disponíveis no TK namespace a partir de qualquer código dentro do projeto. Os métodos disponíveis são definidos em http://www.spinengenharia.com.br/help/an-2014/runtime/index.html

Teclados personalizados em Telas

Ao usar painéis touchscreen, você pode personalizar um teclado já fornecido no Action.NET sobre tela. Ou pode fazer uma tela-Teclado própria.. Contate-nos para exemplos de código fonte sobre a implementação de teclados personalizados.

Toolkit de controles WPF

Você pode adicionar qualquer controle WPF diretamente em telas do Action.NET sem necessidade de quaisquer ferramentas adicionais. Mas, se você quer ter um diálogo de configuração personalizada, acessando os Tags na aplicação, você pode usar as interfaces do WK (Toolkit de controles WPF). Contate-nos para obter exemplos sobre a criação de controles WPF, ou se você precisar de ajuda para encapsular controles Active-X ou dot NET para usá-los dentro de telas Action.NET.

API de acesso a dados COM

Se você precisar acessar o banco de dados em tempo real a partir de aplicações externas, usando qualquer linguagem de programação ou mesmo código JavaScript em navegadores web, você pode usar a API de acesso a dados COM, descrita em http://www.spinengenharia.com.br/help/an-2014/runtime/index.html

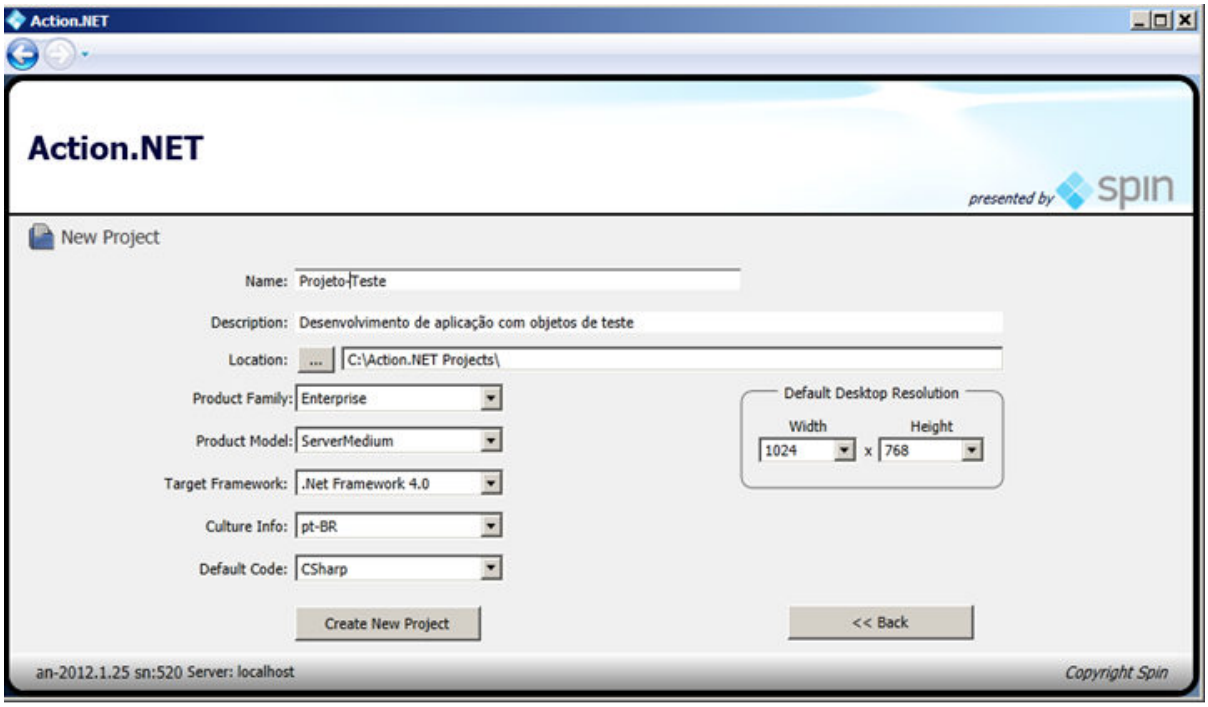
Assistente de Importação e Toolkit de drivers

Action.NET tem extensões para permitir que você crie seus próprios "drivers" de comunicação com dispositivos (protocolos de comunicação), usando o Toolkit de drivers, bem como Ferramentas de importação personalizadas e componentes "add-in". Contate SPIN se você precisar de informações sobre essas extensões adicionais.

Aplicação Default Enterprise

Objetivo

Quando o usuário cria uma nova aplicação a partir da aplicação Enterprise, como apresentado na figura abaixo, esta já vem com um conjunto de tags, telas, alarmes, etc. pré-prontos. Estas informações são designadas "Aplicação Default" e permitem que o usuário às utilize para mais rapidamente gerar a sua aplicação.



A seguir são apresentadas as informações sobre a "Aplicação Default" e como usá-las no desenvolvimento de uma nova aplicação.

Deve-se observar que as informações apresentadas aqui deverão evoluir com o tempo, dado que a Spin, em função de solicitação de usuários, deverá sempre aprimorar a Aplicação Default, disponibilizando novas funcionalidades, Símbolos, Figuras, etc.

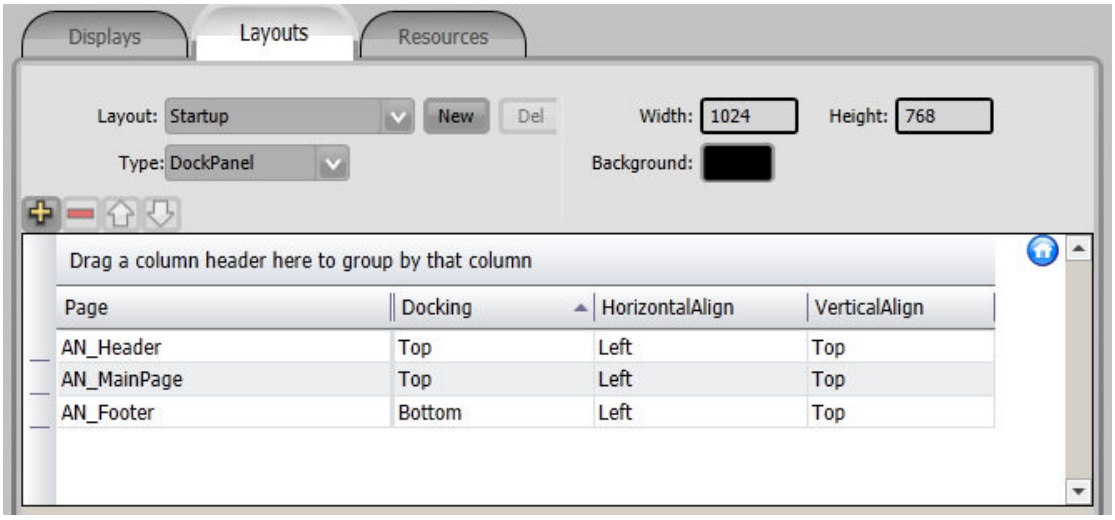
Objetos da Aplicação Default

A Aplicação Default já vem com um conjunto de telas prontas que o usuário poderá usar em sua aplicação. Para o tratamento destas tela também existe um conjunto de **tags,** pré-definidos.

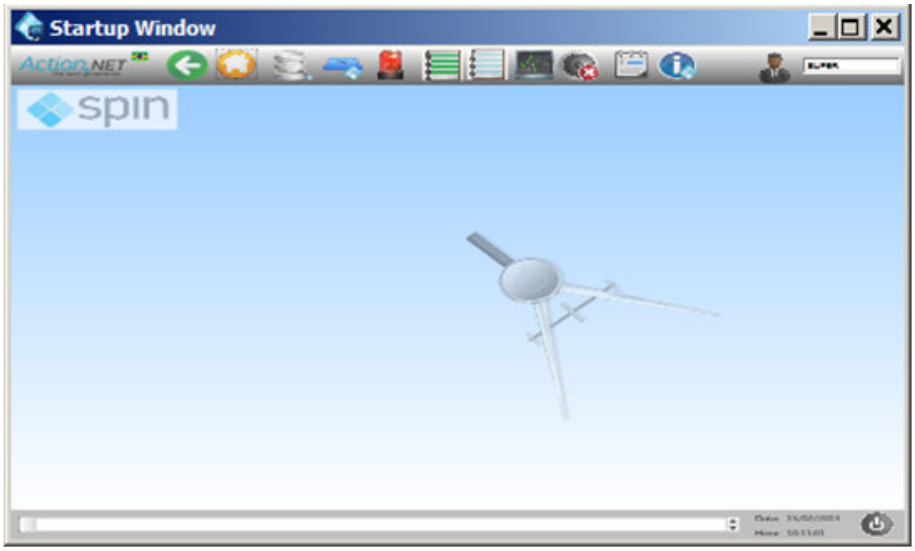
Nesta seção são apresentados estes objetos.

Layout Default

Existe um único layout na Aplicação Default: o layout **Startup** que tem a forma de uma tela com cabeçalho e rodapé, conforme a configuração mostrada na figura abaixo:



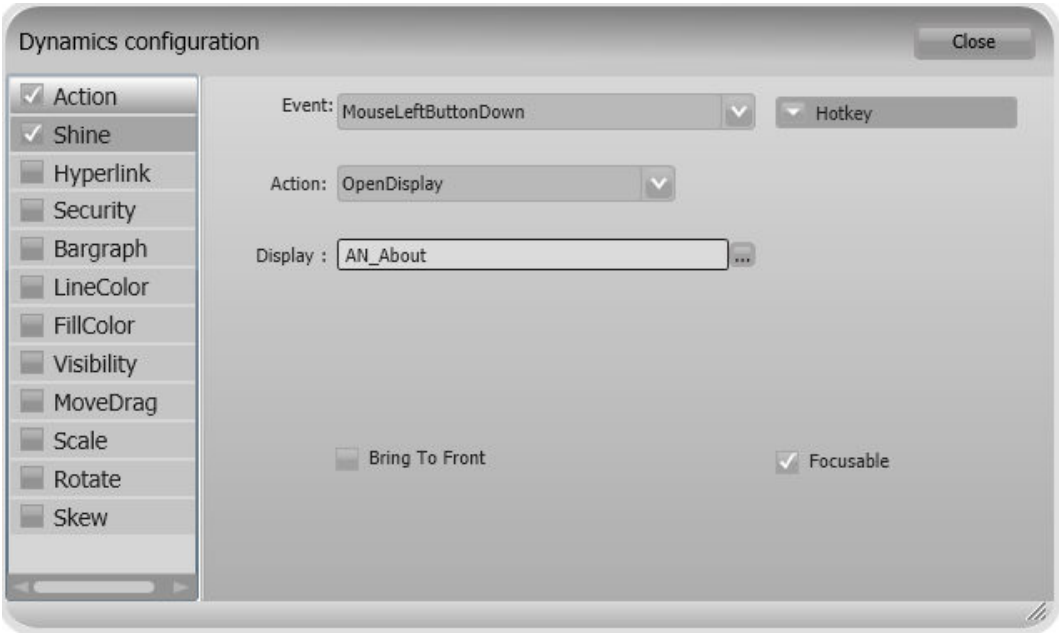
O formato geral da tela que utiliza este layout Startup, é mostrado na figura seguinte:



Este layouts é usado para apresentar todas as telas prontas da Aplicação Default.

Caso os usuários desejam desenvolver novos layouts, basta criá-los e observar que antes de carregar uma tela deve-se carregar seu leiaute.

Abaixo, a título de exemplo, é mostrada a dinâmica do botão que abre a tela de informações, onde se executa duas expressões que abrem, respectivamente, o leiaute e a tela de informações:



Telas Disponibilizadas

Acima é apresentado o cabeçalho da aplicação default onde:

O símbolo quando selecionado, troca o idioma da aplicação default (inglês ←→/ português);

O símbolo retorna para a tela anterior;

O símbolo vai para a tela principal da aplicação default, que inicialmente é uma tela vazia com um compasso. O usuário pode alterar esta tela (AN_MainPage) para ser sua tela de abertura;

O símbolo abre a tela de consulta a medidas históricas (AN_HistoricMeasuresQuery);

O simbolo abre a tela de consulta ao sumário de tags da aplicação (AN_TagsSummary);

O simbolo abre a tela de alarmes correntes (AN_AlarmsSummary);

O símbolo abre a tela de eventos do dia (AN_EventsSummary);

O símbolo abre a tela de consulta a eventos históricos (AN_HistoricEventsSummary);

O simbolo abre a tela de tendência em tempo real e histórica (AN_TrendScreen);

O símbolo abre a tela de log de operação (AN_Operation_Log);

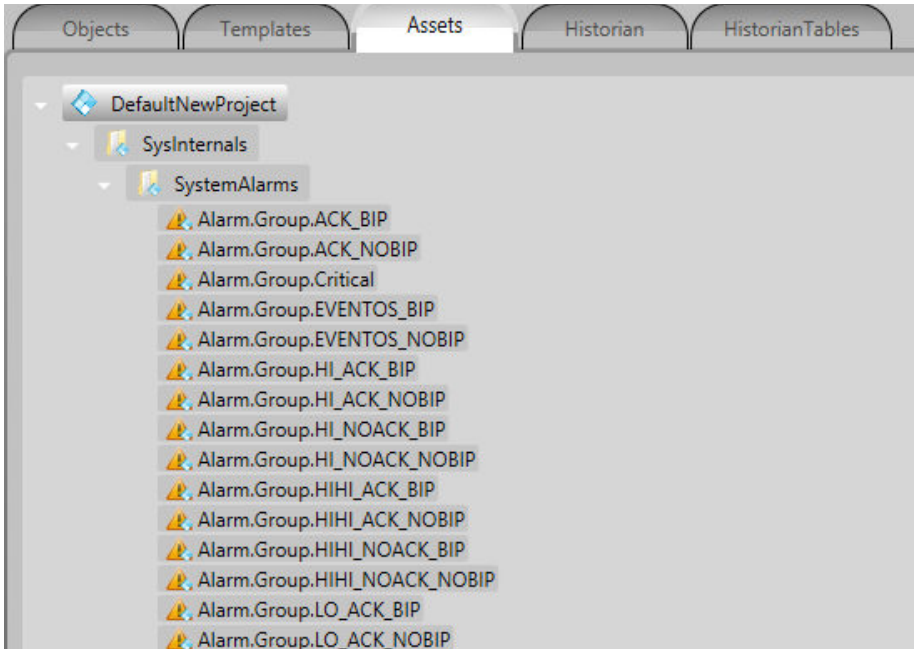
O símbolo cala o alarme sonoro;

O símbolo abre a tela de informações do projeto (AN_About).

Quando a Aplicação Default é ativada, estas telas estão vazias e sem filtros, pois não existem variáveis do usuário nesta aplicação.

Criação dos Níveis (Assets)

Todos os tags existentes na "Aplicação Default" foram criados para gerar as telas disponibilizadas nesta aplicação assim como fazer filtros destas telas que permitem selecionar coleções de tags associados à prioridade de alarme, categoria, nível, etc.

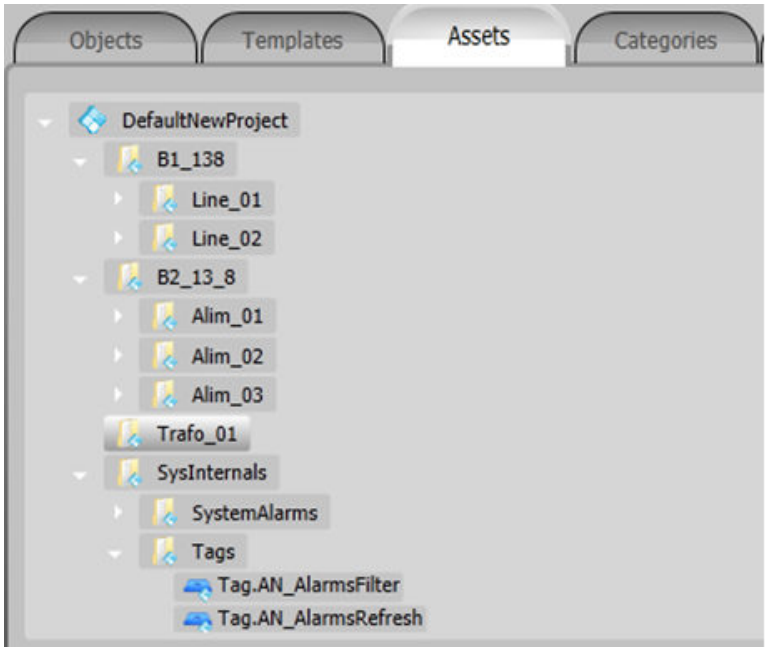


Estes tags são associados ao nível "**SysInternals**" que corresponde a um conjunto de variáveis internas da aplicação, não utilizáveis nas telas geradas automaticamente. Dessa forma, as variáveis utilizadas nestas telas têm como restrições:

- Não podem ser do tipo SysInternals;
- Devem ser de nível (Asset) similar ao SysInternal ou superior;
- Devem ser do domínio do servidor (isto é, não podem ser variáveis associadas a uma estação de trabalho = Client);

Antes do usuário criar seus templates e tags, ele deve fazer a árvore da sua aplicação (Assets) que corresponderá aos níveis utilizados para filtrar variáveis.

Como exemplo, abaixo é apresentada uma árvore de níveis considerando uma subestação com duas linhas de 138 kV uma barra B1, um transformador T1 de 138/13,8 kV e três bays de alimentadores de 13,8 kV.



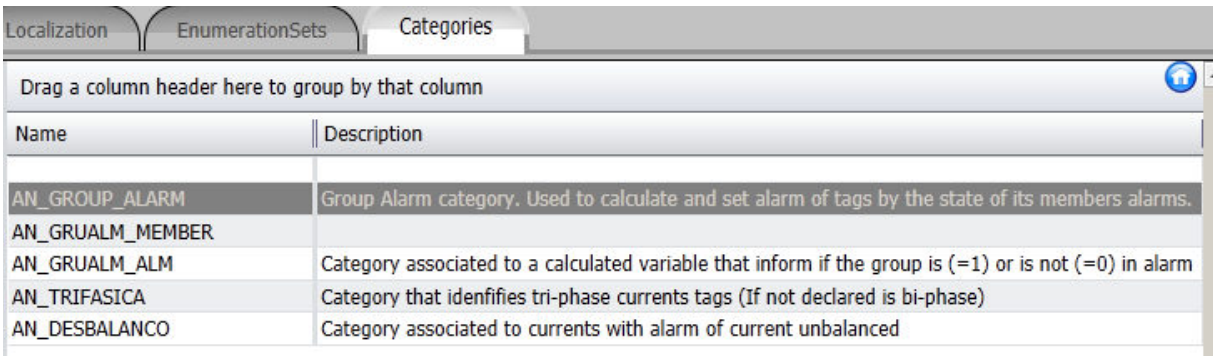
No caso acima, serão criados seis bays (vãos):

- Line_01
- Line_02
- Trafo_01
- Alim_01
- Alim_02
- Alim_03

Categorias

Na "Aplicação Default" são disponibilizadas quatro categorias que tem as seguintes funcionalidades:

1. AN_GROUP_ALARM: Esta categoria deve ser associada a todo o bay / agrupamento ao qual se deseja associar um alarme de grupo. Assim, por exemplo, se a cada alimentador deseja-se criar uma variável que informa se existe algum tag deste alimentador em alarme, deve-se associar este alimentador a esta categoria;
2. AN_GRUALM_ALM: Todos os bays / agrupamentos que tem tratamento de alarme de grupo deverão ter uma variável calculada do tipo digital com esta categoria associada. Esta variável terá o valor igual a 1, se pelo menos um tag deste bay / agrupamento estiver em alarme;
3. AN_DESBALANCO: Esta categoria é utilizada para o cálculo de desbalanço de corrente de circuitos com duas e três fases. Circuitos com três fases também deverão incluir a categoria AN_TRIFASICA. Observar que no calculo feito na Aplicação Default pela rotina: (Script / Classes / AN_DesbalançoCorrente) é exigido que as variáveis sejam declaradas no template Correntes.
4. AN_TRIFASICA: Categoria usada em desbalanço de corrente trifásica.



Security da Aplicação Default

Na aplicação Default estão criados sete usuários com as permissões conforme a figura abaixo:

As senhas são:

Gest → sem senha Administrador → sem senha

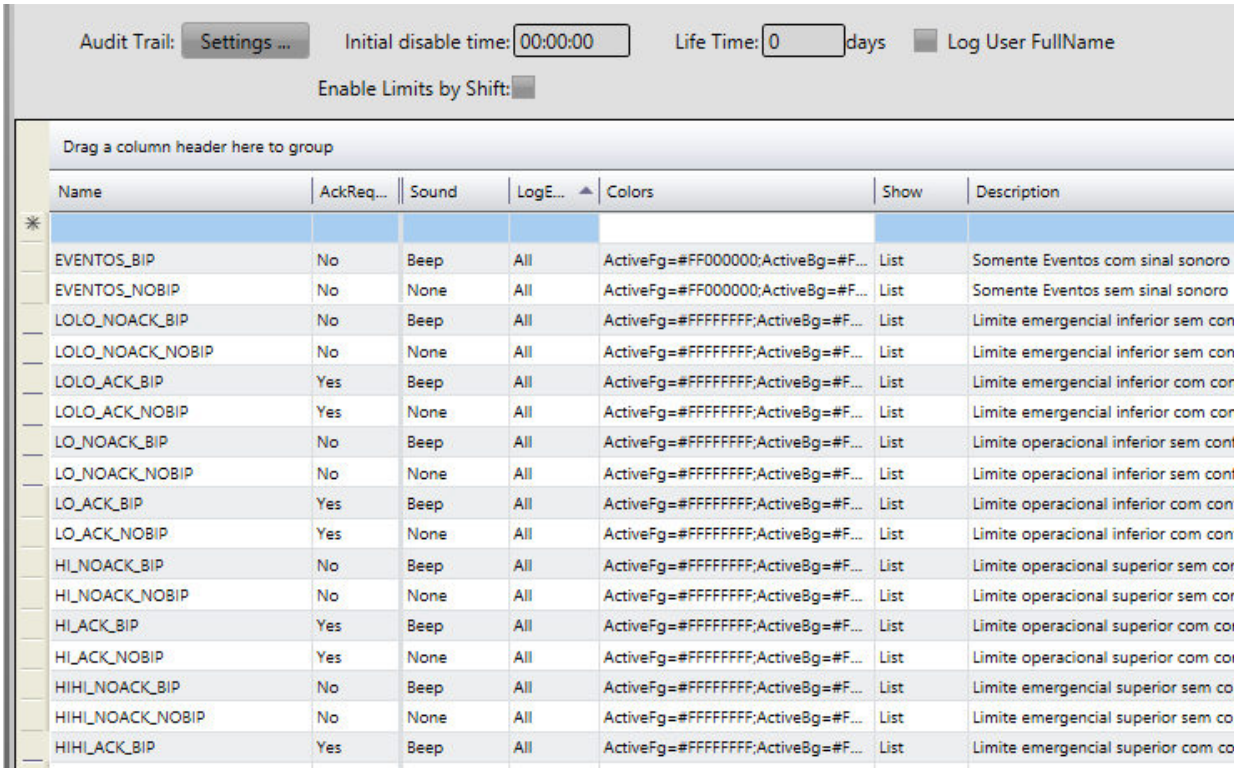
User → sem senha Super → s

Admin → sdop3985 Opercom → o

Oper → o

Alarmes da Aplicação Default

Na Aplicação Default já estão criados vários comportamentos de Alarmes com procedimentos padronizados em relação a sua inserção nas telas geradas automaticamente. A figura abaixo mostra estes alarmes:



Na Aplicação Default as mensagens geradas pelos AlarmGroups que tiverem prioridades entre 1 e 10, serão inseridas no **Sumário de Alarmes,** e somente permanecerão lá enquanto estiverem no estado **ACTIVE.** Todas as mensagens, tanto de ativação quanto de normalização de alarmes são inseridas no **Sumário de Eventos** e permanecem lá durante todo o dia em que foram geradas.

Portanto a definição de prioridade deverá ser feita pelo usuário de acordo com sua preferência no tratamento das mensagens.

No ambito do Action.NET, denomina-se **Eventos** todas as ocorrências que devem ser registradas em histórico e denomina-se **Alarmes** as ocorrências que devem ser verificadas e requerem uma ação, pois se referem a situações de falha na aplicação.

Os textos utilizados para compor os nomes dos AlarmGroups tem o seguinte significado:

ACK - Exigem reconhecimento

NOACK = Não exigem reconhecimento

BIP - Causam alarme sonoro

NOBIP - Não utilizam alarme sonoro na sua ativação

Utilizam **condition Change** e são destinados a somente aparecer em Sumario de Eventos, Os itens devem ser criados com prioridade maior do que 10.

EVENTOS_BIP

EVENTOS_NOBIP

Utilizam **condition HI**

HI_ACK_BIP

HI_ACK_NOBIP

HI_NOACK_BIP

HI_NOACK_NOBIP

Utilizam **condition HIHI**

HIHI_ACK_BIP

HIHI_ACK_NOBIP

HIHI_NOACK_BIP

HIHI_NOACK_NOBIP

Utilizam **condition LO**

LO_ACK_BIP

LO_ACK_NOBIP

LO_NOACK_BIP

LO_NOACK_NOBIP

Utilizam **condition LOLO**

LOLO_ACK_BIP

LOLO_ACK_NOBIP

LOLO_NOACK_BIP

LOLO_NOACK_NOBIP

Utilizam **condition Equal ou NotEqual**

ACK_BIP

ACK_NOBIP

NOACK_BIP

NOACK_NOBIP

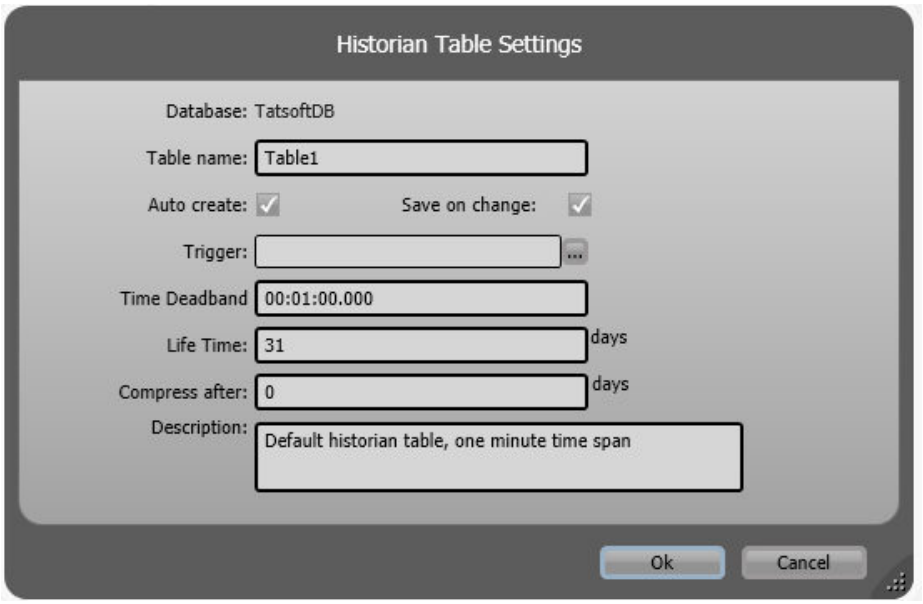
NOMSG_BIP

NOMSG_NOBIP

Tabelas Historian

Na Aplicação default existe uma tabela já criada que pode ser usada para armazenar medidas históricas. Esta tabela chama-se Table1 e tem os seguintes atributos:

- Table Name: Table1;
- Auto Create: sim;
- Save on Change: sim (gravará um registro sempre que um Tag da tabela for alterado);
- Trigger: nenhum;
- Life Time: 31 dias (apaga os registros mais antigos);
- Time Span: 1 minuto;
- Description: Default historian table, one minute time span.



Criando aplicação a partir da Default

Este capitulo seção apresenta passo a passo a elaboração de um projeto partindo da Aplicação Default. Os tópicos a seguir apresentados são:

- "Criação dos Dicionários" na pagina 235
- "Criação dos Templates" na pagina 236
- "Criação de Níveis (Assets)" na pagina 239
- "Criação dos tags de Bays" na pagina 240
- "Criação dos Devices" na pagina 241
- "Criação do Nodo" na pagina 242
- "Criação dos Alarmes" na pagina 243
- "Criação do Histórico de Medidas" na pagina 244
- "Criação das Telas da Aplicação" na pagina 245

Criação dos Dicionários

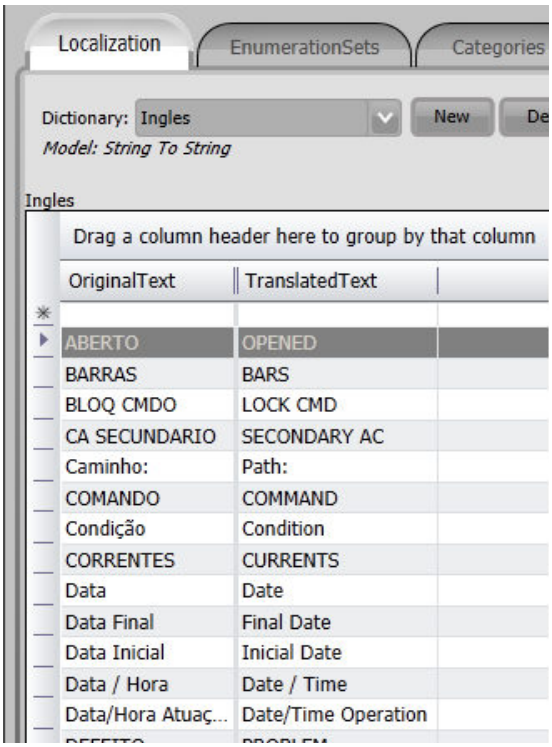
Na criação de tags e templates, deve-se definir o dicionário que será usado para traduzir o valor de uma variável para seu estado. Este atributo chamado **Parameter** é usado nas listas de Alarme. Assim, se o tag de uma proteção vale "1", na lista de alarme será apresentado o texto "ATUADO".

Deve-se assim, criar os dicionários que serão usados para posteriormente criar os templates e tags.

Foram criados os dicionários:

- SEL_CHAVE: Estados de chaves de 4 posições;
- SEL_PROTECTION: Digitais com frase Normal / Atuado;
- DNP_IIN_CMD: Resultado dos comandos DNP (Internal Indication);

Se o usuário deseja manter a aplicação em duas línguas, deverá:



1. Incluir no dicionário Inglês, apresentado na figura abaixo, todos os termos em português que colocar em telas e objetos.
2. Em todos os objetos com texto usar a propriedade **Localization;**
3. Em dinâmicas com texto e em Scripts, no local de textos sempre usar `@Client.Locale("texto")` e adicionar o texto ao dicionário Inglês;

Criação dos Templates

Introdução

Na organização de uma nova aplicação, o usuário deve planejar adequadamente os templates. Assim, a título de exemplo, vai-se criar uma aplicação de monitoração de uma subestação com:

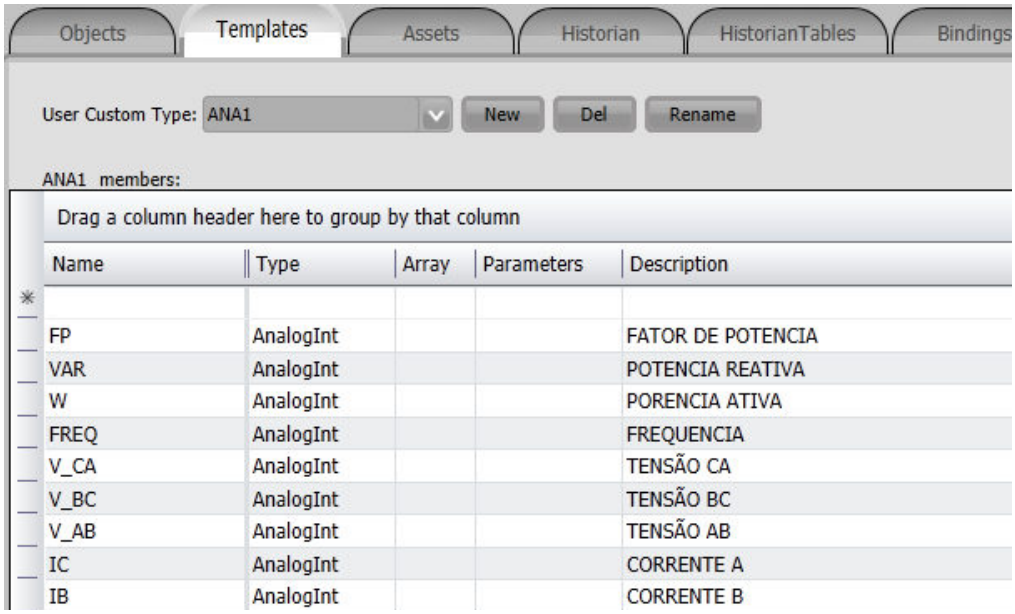
- Bay de linha;
- Bay de transformador;
- Bay de alimentador;

Como um bay pode ocorrer mais de uma vez, para cada bay será criado como um template.

Os pontos de cada bay são fornecidos, geralmente, em planilhas Excel e, a partir destas planilhas, deverão ser criados os templates, isto é, é mais fácil adequar à planilha Excel para o formato do ActionNET do que digitar ponto por ponto diretamente nos Tags e Templates.

Estudando os tags de um bay o usuário poderá observar sequências de pontos que se repetem em diferentes bays. Estas sequências, a título de facilitar o trabalho e padronizar tags, devem ser também

criadas como templates. Assim, por exemplo, em diferentes bays se tem a mesma sequência de variáveis analógicas e, para facilitar o trabalho, define-se esta sequência como um template, conforme a figura abaixo:



Name	Type	Array	Parameters	Description
FP	AnalogInt			FATOR DE POTENCIA
VAR	AnalogInt			POTENCIA REATIVA
W	AnalogInt			PORENCIA ATIVA
FREQ	AnalogInt			FREQUENCIA
V_CA	AnalogInt			TENSÃO CA
V_BC	AnalogInt			TENSÃO BC
V_AB	AnalogInt			TENSÃO AB
IC	AnalogInt			CORRENTE A
IB	AnalogInt			CORRENTE B

Template dos Bays

Para gerar o template de um bay, a partir de uma planilha Excel, basta organizar seus pontos de forma idêntica a da Aba de templates. Para criar uma planilha organizada de forma idêntica a Aba de Templates, basta copiar uma linha desta Aba e colá-la em uma planilha Excel, conforme a planilha abaixo gerada a partir do template *ANA1*.

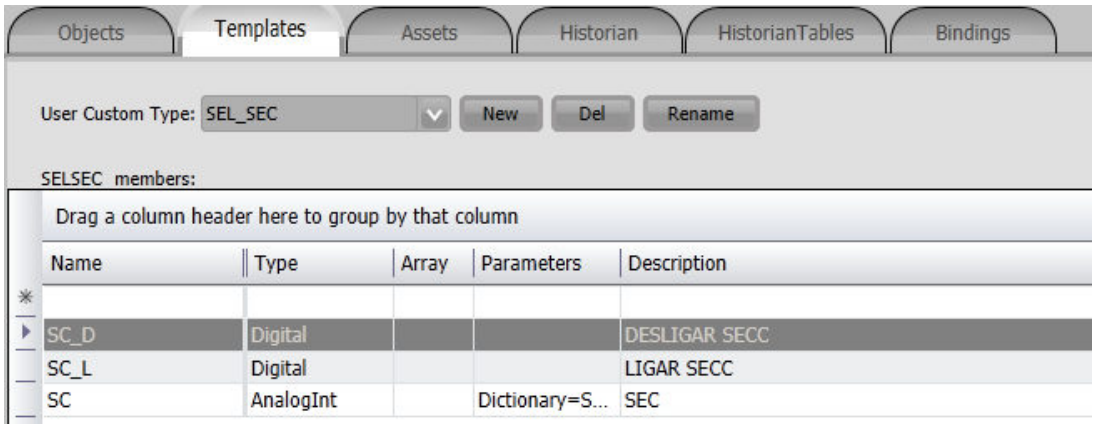
Name	Type	Array	Parameters	Min	Max	EngUnits	Format	StartValue	Retentive	Visibility	Comment	Category	Description
FP	AnalogDouble								Properties	Protected			FATOR DE POTENCIA
VAR	AnalogDouble								Properties	Protected			POTENCIA REATIVA
W	AnalogDouble								Properties	Protected			PORENCIA ATIVA
FREQ	AnalogDouble								Properties	Protected			FREQUENCIA
V_CA	AnalogDouble								Properties	Protected			TENSÃO CA
V_BC	AnalogDouble								Properties	Protected			TENSÃO BC
V_AB	AnalogDouble								Properties	Protected			TENSÃO AB
IC	AnalogDouble								Properties	Protected			CORRENTE A
IB	AnalogDouble								Properties	Protected			CORRENTE B
IA	AnalogDouble								Properties	Protected			CORRENTE A

A partir desta planilha, esvaziam-se as linhas, mantendo o cabeçalho, e depois se deve organizar nossa planilha de pontos dos bays, no formato desta planilha, conforme mostrado na figura abaixo, onde foram colocados os pontos do bay de linha existentes em um relé SEL_311.

	Name	Type	Array	Parameters	Min	Max	EngUnits	Format	StartValue	Retentive	Visibility	Comment	Category	Description
2	SV1	Digital		Dictionary=SEL_PROTECTION;						Properties	Protected			Fase A
3	SV2	Digital		Dictionary=SEL_PROTECTION;						Properties	Protected			Fase B
4	SV3	Digital		Dictionary=SEL_PROTECTION;						Properties	Protected			Fase C
5	SV4	Digital		Dictionary=SEL_PROTECTION;						Properties	Protected			Zona 1
6	SV5	Digital		Dictionary=SEL_PROTECTION;						Properties	Protected			Zona 2
7	SV6	Digital		Dictionary=SEL_PROTECTION;						Properties	Protected			Zona 3
8	SV7T	Digital		Dictionary=SEL_PROTECTION;						Properties	Protected			59N
9	IN104	Digital		Dictionary=SEL_PROTECTION;						Properties	Protected			Recepção Bloq. Seletividade Lógica
10	IN105	Digital		Dictionary=SEL_PROTECTION;						Properties	Protected			Bx. Pressão SF6
11	P_81D1T	Digital		Dictionary=SEL_PROTECTION;						Properties	Protected			81 Estágio 1 Freq.
12	P_81D2T	Digital		Dictionary=SEL_PROTECTION;						Properties	Protected			81 Estágio 2 Freq.
13	RMB1A	Digital		Dictionary=SEL_PROTECTION;						Properties	Protected			Recepção TDD
14	RBADA	Digital		Dictionary=SEL_PROTECTION;						Properties	Protected			Falha Teleprot.
15	P_51PT	Digital		Dictionary=SEL_PROTECTION;						Properties	Protected			51PT Sobrecorrente Temporizado Fase
16	P_51GT	Digital		Dictionary=SEL_PROTECTION;						Properties	Protected			51GT Sobrecorrente Temporizado Neutro
17	P_67G1T	Digital		Dictionary=SEL_PROTECTION;						Properties	Protected			67G1T Sobrecorrente Inst. Neutro
18	P_67P1T	Digital		Dictionary=SEL_PROTECTION;						Properties	Protected			67P1T Sobrecorrente Inst. Fase
19	LOP	Digital		Dictionary=SEL_PROTECTION;						Properties	Protected			Perda de Potencial
20	RB5	Digital								Properties	Protected			Reset
21	RB6	Digital								Properties	Protected			CMD L/R
22	DJ_52A	AnalogInt		Dictionary=AN_DISJ_STATES;						Properties	Protected			DISJ 52A
23	IN102	AnalogInt		Dictionary=SEL_CHAVE;						Properties	Protected			SC 89-1
24	IN103	AnalogInt		Dictionary=SEL_CHAVE;						Properties	Protected			SECC 89- 2
25	IIN_VAL	AnalogInt								None	Protected			Internal Indication
26	IIN_CMD	AnalogInt								None	Protected			Status dos Comandos
27	CB	SEL_DJ												DISJ 52A
28	SC_89-1	SEL_SEC												SC 89-1
29	ALM_GROUP	Digital		Dictionary=SEL_PROTECTION;						None	Protected		AN_GRUALM_ALM	Alarme de grupo
30	ANA	ANA1								None	Protected			

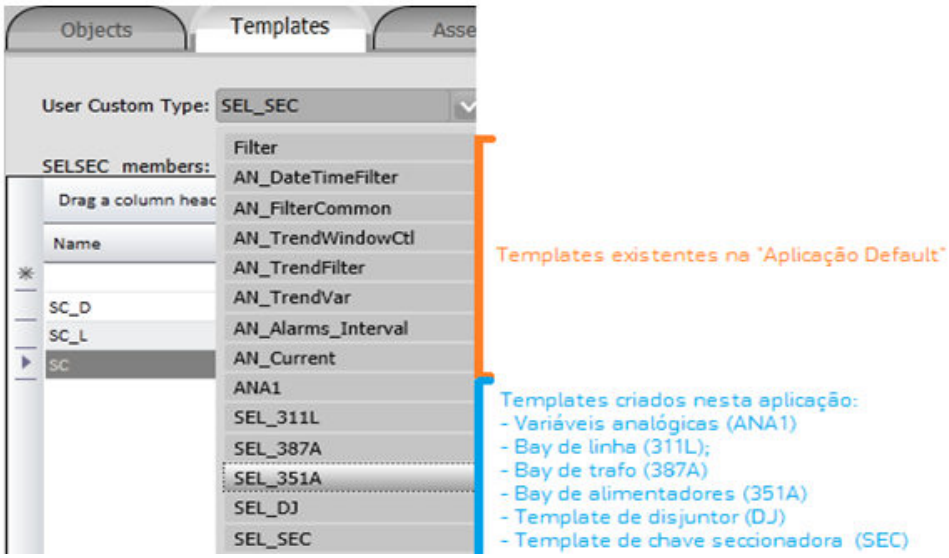
Nesta planilha, as células em azul são os pontos originais da planilha de pontos do bay de linha e os pontos em branco correspondem a pontos criados pelo usuário, pelos seguintes motivos:

1. ANA: corresponde aos pontos analógicos do bay que estão apresentados na figura do item 8.2.1.
2. ALM_GROUP: é uma variável calculada que deve ser colocada em todos os bays onde se deseja ter um alarme de grupo identificando se este bay tem ou não pelo menos um alarme atuado. Observar que em
3. Em **Category** se tem a variável AN_GRUALM_ALM associada.
4. IIN_VAL e IIN_CMD: são variáveis associadas ao canal DNP3 que serão explicadas mais a frente, quando da criação dos pontos de Entrada e Saída da aplicação.
5. CB (Circuit Break): foi criado um template para o disjuntor onde estão declaradas a variável de entrada com o estado do disjuntor e as duas variáveis de saída para ligar / desligar o disjuntor;
6. SC_89-1: Foi criado um template para chaves seccionadoras onde estão declaradas a variável de entrada com o estado da chave e as duas variáveis de saída para ligar / desligar a chave;



As colunas do template usadas são:

- Type: Tipo do ponto que no caso são:
  - Digital (0 ou 1): variáveis digitais de entrada e saída;
  - AnalogInt: Variáveis digitais discretas são definidas como AnalogInt: estado de disjuntor e seccionadoras (assumem valores entre 0 e 3).
  - AnalogDouble: Variáveis analógicas são definidas como AnalogDouble.
  - ANA1: template com todas as variáveis analógicas de um bay.
- Parameter: contém o nome do dicionário que converte o valor da variável em um texto;
- Retentive: o texto properties indica que no início do programa tanto o valor do Tag como suas propriedades serão iguais aos seus valores quando do encerramento do programa.
- Category: o item "7.3.2" trata as categorias existentes na aplicação default e, no caso, criou-se um único tag referente ao alarme de grupo associado a este bay.
- Description: texto descritivo do tag.



Assim como se criou o bay associado ao relé SEL_311L (proteção de linha) se deve criar os pontos associados aos relés que coletam pontos dos alimentadores e do transformador.

Criação de Níveis (Assets)

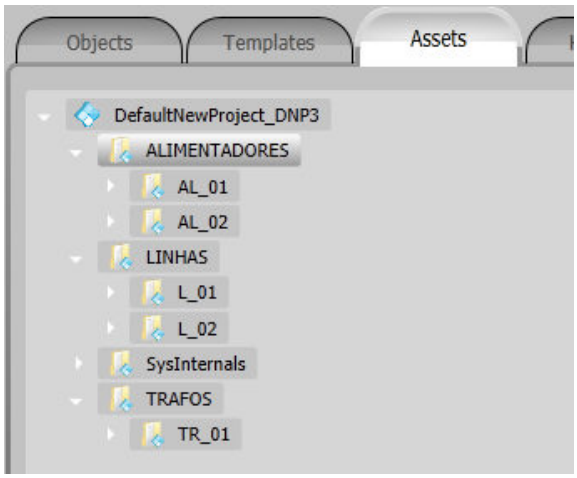
Os Assets (níveis) correspondem à organização dos agrupamentos de pontos da aplicação. Os tags tratados a partir da aplicação default devem ser diferentes de *SysInternals* que é o agrupamento dos tags usados nas rotinas internas da aplicação Default, para gerar as telas com os diversos relatórios disponibilizados automaticamente:

- Histórico de Medidas;

- Sumário de Tags;
- Sumário de Alarmes;
- Sumário de Eventos;
- Pesquisa a Eventos;
- Log de Operação;
- Tendência em Tempo Real e Histórica;

Usando **NewLevel** (botão direito do mouse sobre DefaultNewProject_DNP) definiu-se, conforme figura a seguir, três novos itens de segundo nível, conforme mostra a figura abaixo:

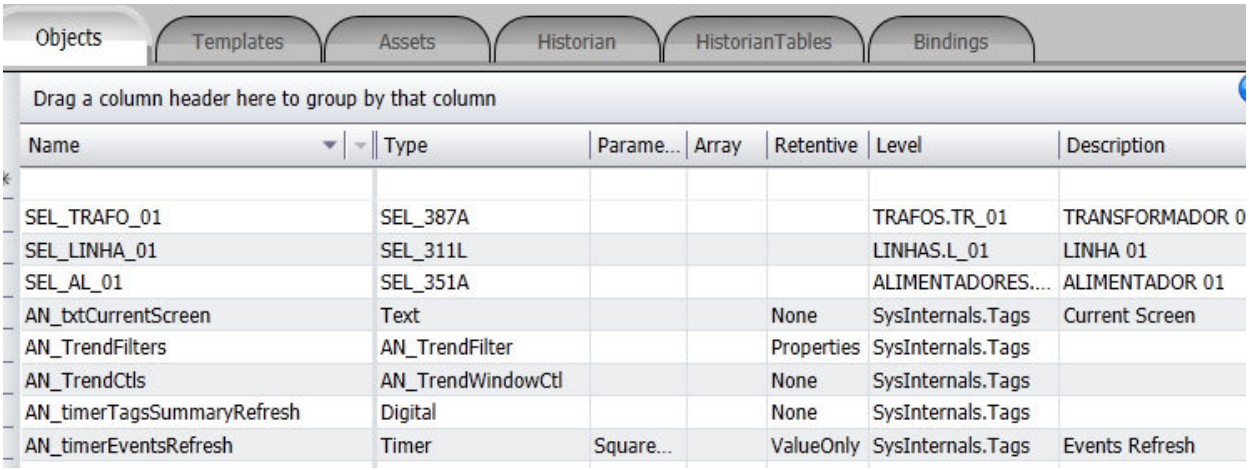
- Alimentadores: terá os alimentadores.
- Linhas: terá os bays de linha;
- Trafos: terá os bays de trafo;



Todos os tags criados deverão estar em um destes níveis.

Criação dos tags de Bays

Para criar os bays da subestação, vai-se para a Aba de Objects e criam-se os bays associando-se seus tipos (Type) aos respectivos templates, conforme a figura abaixo.



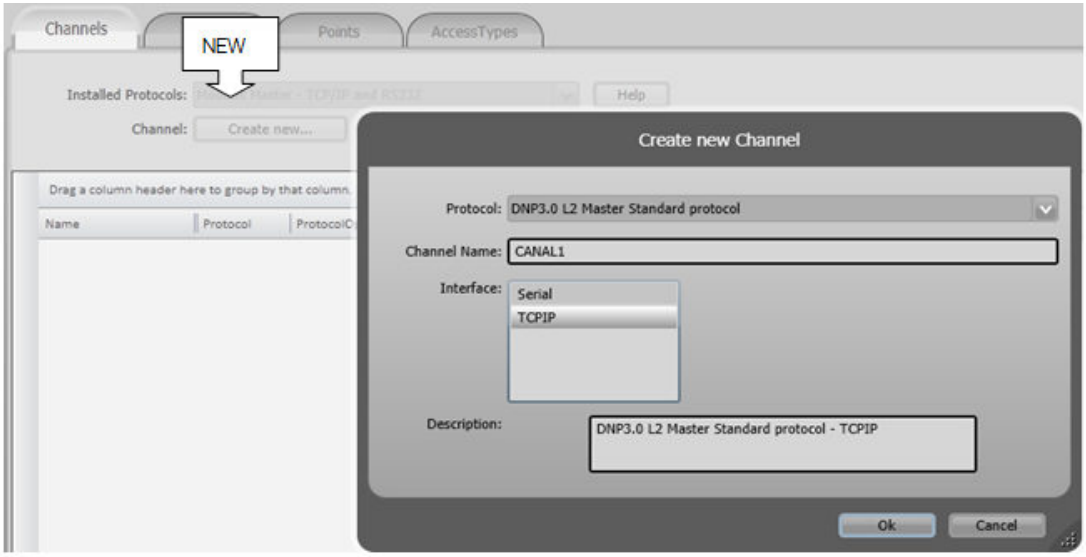
Observar que estes tags devem ser do tipo Server (estão no servidor) e não podem estar incluídos no nível **SysInternal.**

Criação dos Devices

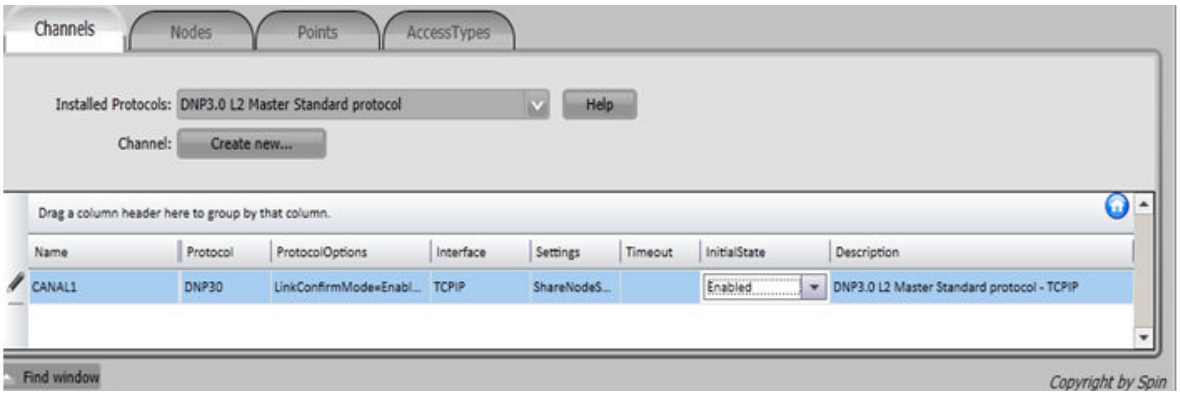
Nesta aplicação exemplo, considera-se que os relés estejam se comunicando com o SCADA através do protocolo DNP 3.0. Para cada protocolo existe um manual específico.

Criação de um Canal DNP3

Na Aba de canais (Channel) cria-se um novo canal chamado "DNP301" usando o protocolo DNP3 com comunicação TCP/IP, conforme figura abaixo:



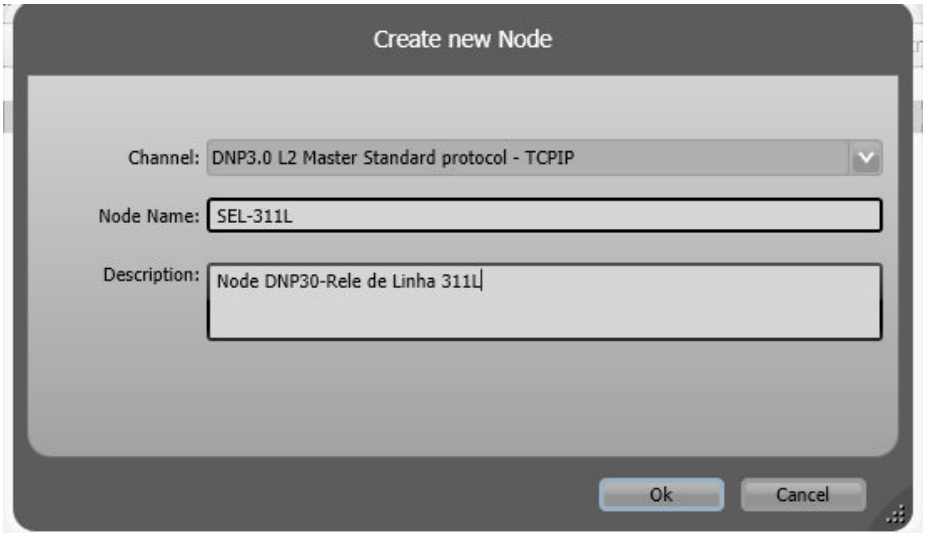
Neste canal se usa os atributos default, conforme abaixo:



Criação do Nodo

O nodo corresponde ao IED (relé, CLP, UTR, etc.).

Na ABA Node deve-se criar um novo nodo conforme a figura abaixo:



Após, deve-se acertar os atributos do node, clicando na coluna Primary Station.

Criação dos Pontos do Canal / Nodo

A criação dos pontos é feita de forma similar a criação dos templates, isto é, a maneira mais fácil é criar um ponto qualquer na ABA de pontos, exporta-lo para uma planilha Excel e após, a partir da planilha de pontos da aplicação criar uma planilha Excel mantendo a organização da planilha exportada:

TagName	Node	Address	DataType	Modifiers	AccessType	Label	Scaling

Tag.SEL_LINHA_01.DJ_52A	Node.SEL_311L_01		Native		AccessType.Read		None

No manual do protocolo DNP 3 existem as informações sobre como preencher cada coluna de um ponto. Os pontos BI são de entrada digital (Binary Input), os pontos AI são entradas analógicas, os pontos CRO são saídas digitais (Control Relay Output) e os pontos IIN são entradas do tipo Internal Indication do DNP3.

Observações:

1. Para criar um ponto duplo como no caso das seccionadoras, definiu-se a seccionadora como uma variável AnalogInt, e criou-se dois pontos digitais que correspondem ao Bit0 e Bit1 da seccionadora (linhas 10 e 11 da planilha acima);
2. As variáveis do tipo IIN (linhas 40 e 41) tem endereço fixo, conforme apresentado no manual do DNP3;
3. Para executar uma saída deve-se chamar a rotina (Class) AN_TAG com método de inverter o valor da variável de saída:

   ```
   @Script.Class.AN_Tags.ToggleDigOutput(<tag de saída>);
   ```

4. Sempre que enviar um comando, é interessante colocar uma mensagem no Sumário de Eventos registrando este comando. Para tal execute a ação:

   `@Alarm.SystemEvents.AddCustomMessage("Mensagem" + txtNome)` onde txtNome é o string com o nome da variável comandada.

Criação dos Alarmes

A criação de alarmes, a partir da Aplicação Default, exige que se crie na tabela de Itens de Alarmes uma entrada (Linha) para cada comportamento que deve ser sinalizado. A tabela abaixo mostra o número de entradas que serão criadas em função da situação:

DESCRIÇÃO	LINHAS	CONDITION	GROUPS	OBSERVAÇÕES
Sinalizar qualquer mudança	1	Changed	EVENTOS_BIP	
Sinalizar diferentemente 0/1 (0)	2	Equal 0	NOACK_BIP	
Sinalizar diferentemente 0/1 (1)	2	Equal 1	NOACK_BIP	
Analógica com 1 Limite (anormal)	1	HI / LO	HI_NOACK_BIP ou LO_NOAC K_BIP	Quando normal – Hora Normalização
Analógica com 2 Limite (HI)	2	HI	HI_NOACK_BIP	Quando normal – Hora Normalização
Analógica com 2 Limite (LOW)	2	LO	LO_NOACK_BIP	Quando normal – Hora Normalização
Analógica com 4 Limites	4	HI	HI_NOACK_BIP	
Analógica com 4 Limites	4	LO	LO_NOACK_BIP	
Analógica com 4 Limites	4	HIHI	HIHI_NOACK_BIP	
Analógica com 4 Limites	4	LOLO	LOLO_NOACK_BIP	

Observações:

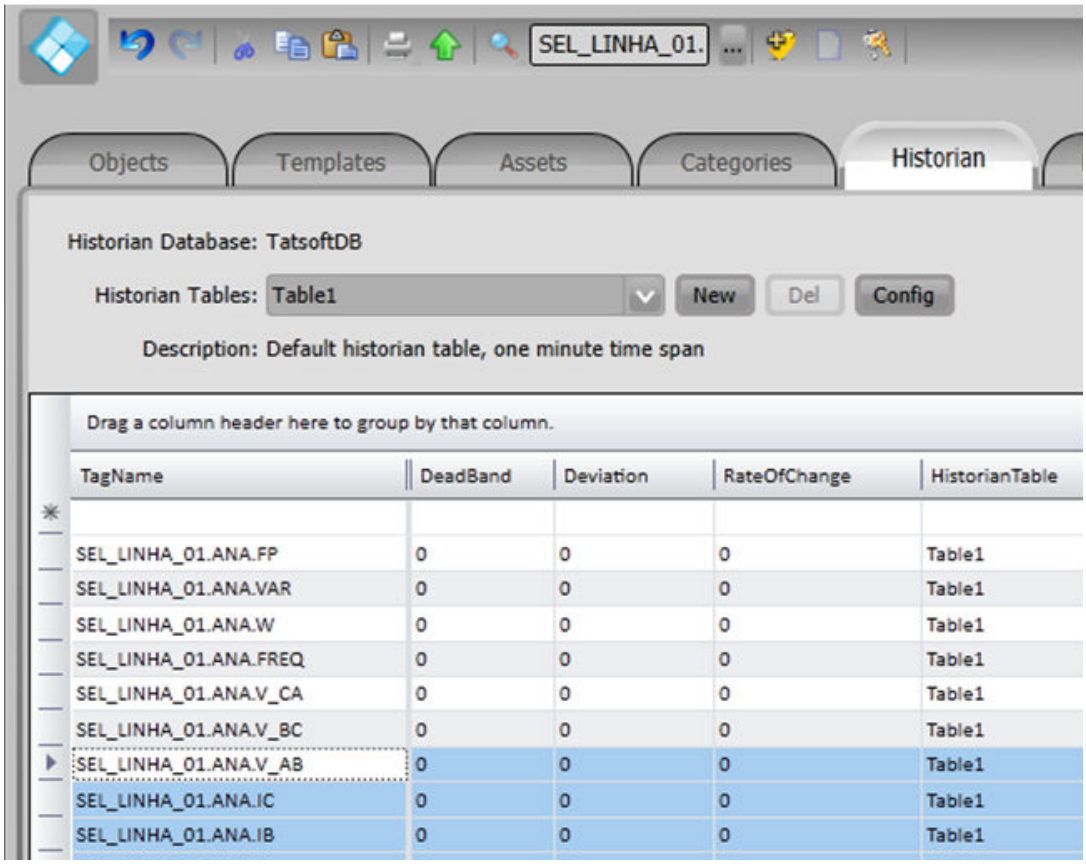
1. Pode-se colocar alarmes de Tags analógicos com dois limites superiores e 2 inferiores.
    - Quando um dos alarmes está atuado, os Tags são mantidos no Sumário de Alarmes e no de Eventos. Quando os Tags são normalizados, eles saem do Sumário de Alarmes e, no Sumário de Eventos é colocado o horário de sua normalização.
    - No ActionNET, quando da definição do Sumário de Alarmes, é selecionado o parâmetro "Merge Hi e HIHI" que mantém na lista o alarme de mais importância quando dois alarmes deste tipo estão atuados, isto é, se HIHI está atuado, a linha de HI é inibida e o mesmo ocorre com LO e LOLO.
2. Há eventos que são armazenados no histórico sempre que ocorre uma mudança de valor. Estes eventos são usados em manobras de equipamentos tais como seccionadora, tap de transformador, etc.

Em tempo de execução, o Sumário de Alarmes apresenta os alarmes que estão atuados no momento.

Já o Sumário de Eventos tem os eventos apresentados em ordem cronológica Eventos analógicos tem seu horário de atuação e normalização sinalizados. Assim, quando um alarme analógico inicia, ele possui apenas o horário de atuação e quando ele encerra, possui os dois horários.

Criação do Histórico de Medidas

Conforme apresentado, na Aplicação Default existe uma tabela (Table1) que é gravada no mínimo a cada minuto e sempre que o tag for alterado. Abaixo é mostrada esta tabela preenchida com todas as variáveis analógicas do bay de linha 01.



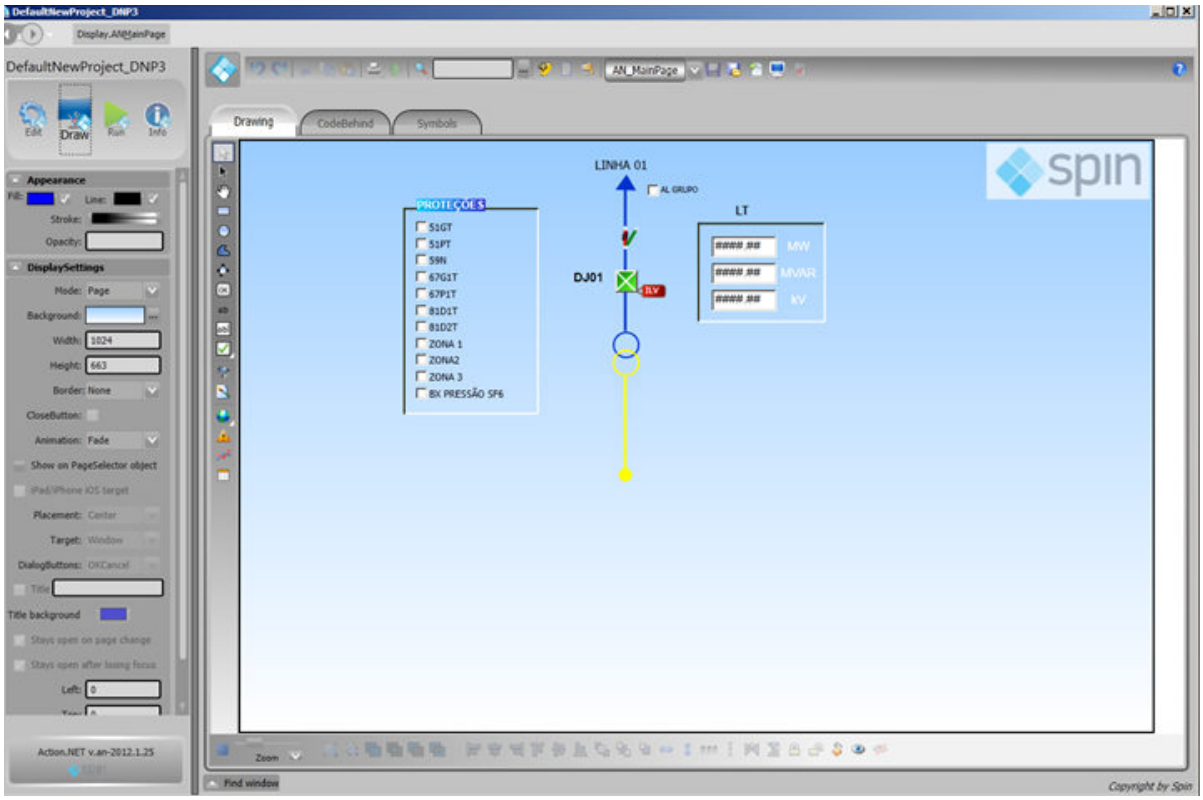
Para cada Bay, deve ser repetido procedimento semelhante.

Criação das Telas da Aplicação

Conforme apresentado no capítulo referente a Aplicação Default, já existem nove telas onde oito já estão prontas, ver em "Telas Disponibilizadas" na página 227.

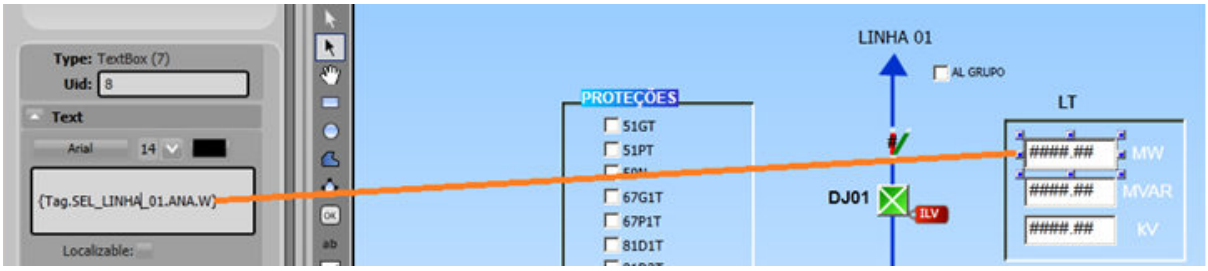
Tela Principal (AN_MainPage)

Na tela principal foram colocadas informações da Linha 01, conforme mostra a figura abaixo:

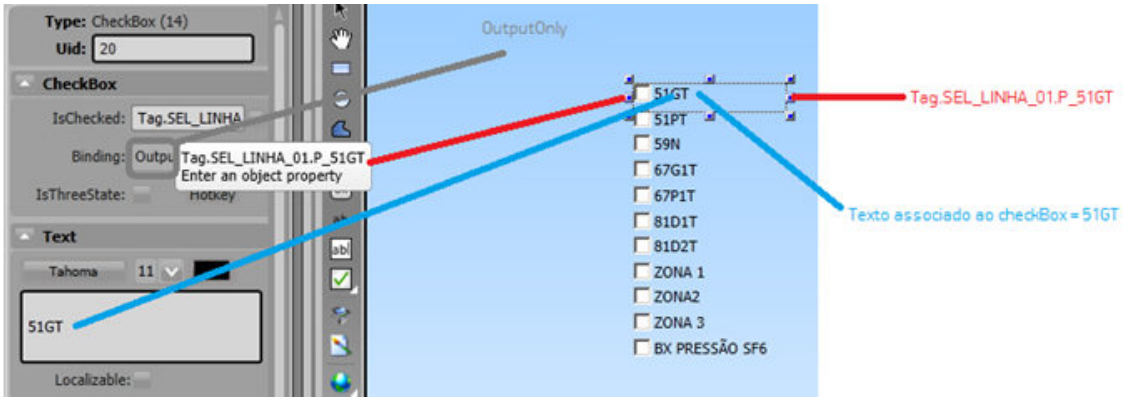


As quatro retas e o texto Proteções são retângulos e TextOutput, respectivamente, que após serem criados foram agrupados, sem nenhum vínculo à tags;

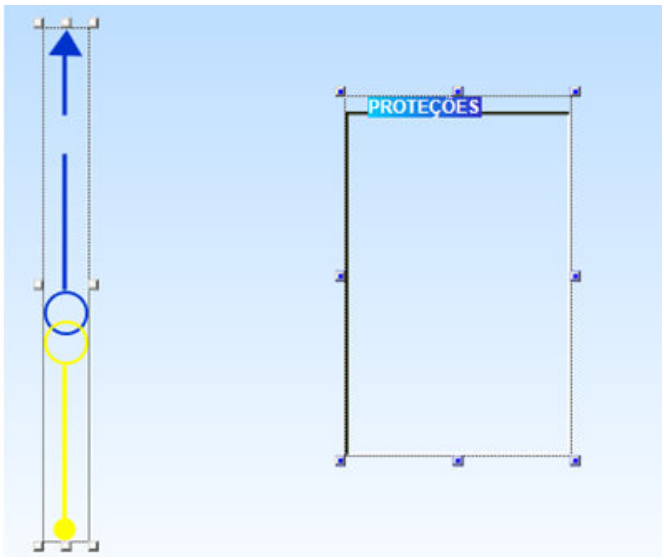
Os valores das três variáveis analógicas a direita são TextBox associados aos tags das variáveis: potência ativa, reativa e tensão.



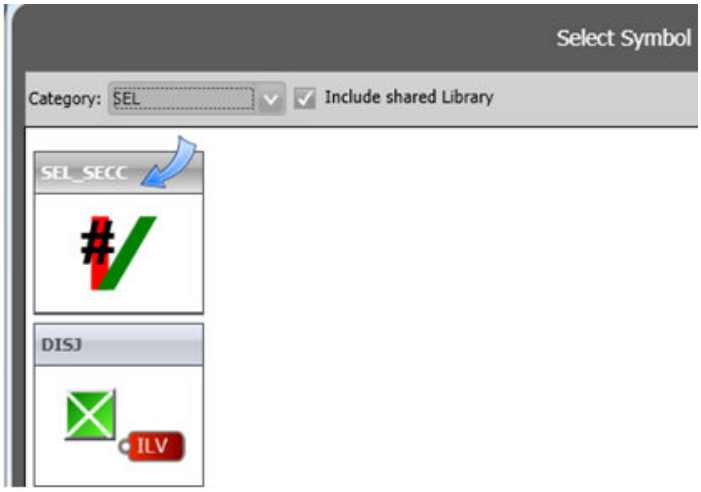
As proteções são checkBox, sendo cada uma associada a um tag do bay de linha. Elas são tipo Output (Binding: Outputonly) o que quer dizer que o usuário nunca faz uma entrada no checkbox;



O desenho do bay de linha foi feito com objetos retângulos, círculos e polígonos (seta) que ao final foram agrupados tornando-se um único objeto:

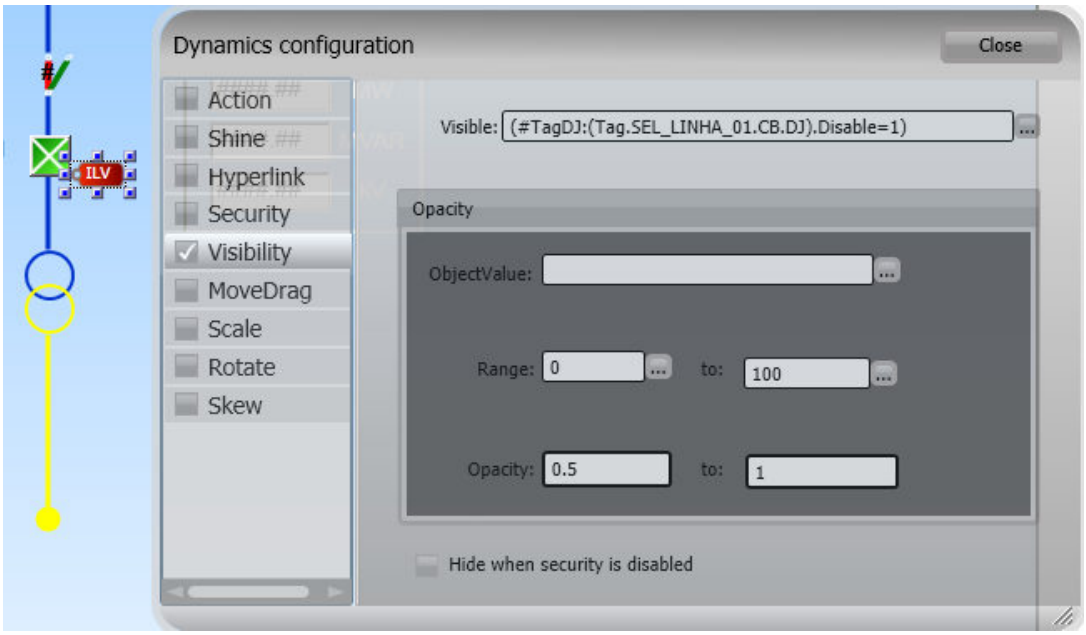


O disjuntor e a seccionadora são Símbolos criados para a aplicação. A seccionadora apresenta a figura de fechado (retângulo vermelho), aberto (reta verde inclinada) e inválido (# assume letra ?).



O disjuntor fica vermelho quando fechado, verde quando aberto e com a bandeirola vermelha ao lado quando impedido. Um clique tanto sobre o disjuntor como a seccionadora abre a tela de comando:

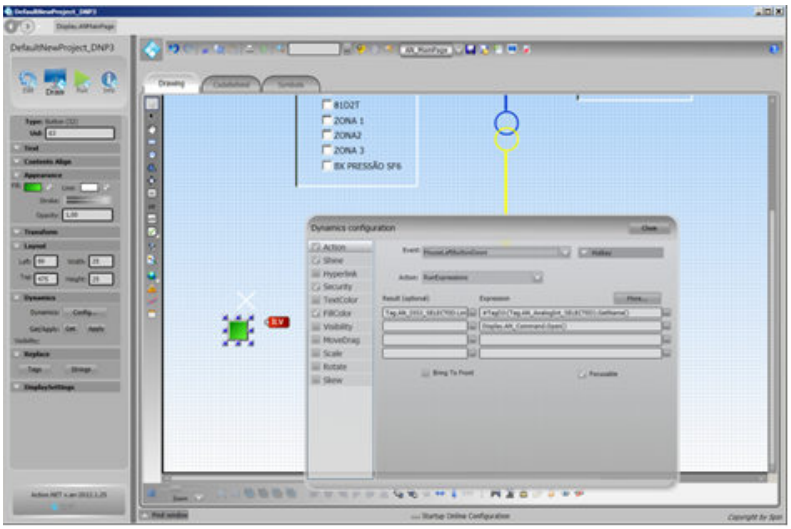
A bandeirola do símbolo DISJ só é visível quando o tag do disjuntor está *Disable*, conforme mostra a dinâmica abaixo:



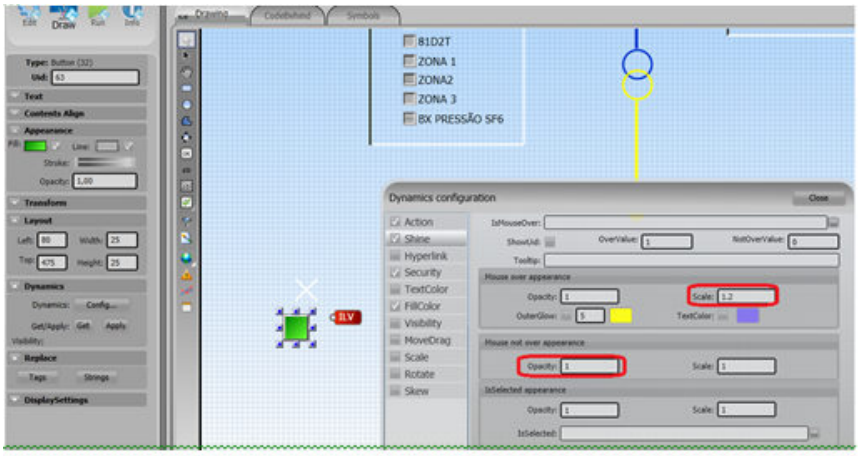
O Disjuntor é um botão com quatro dinâmicas:

Um clique sobre ele faz duas ações:

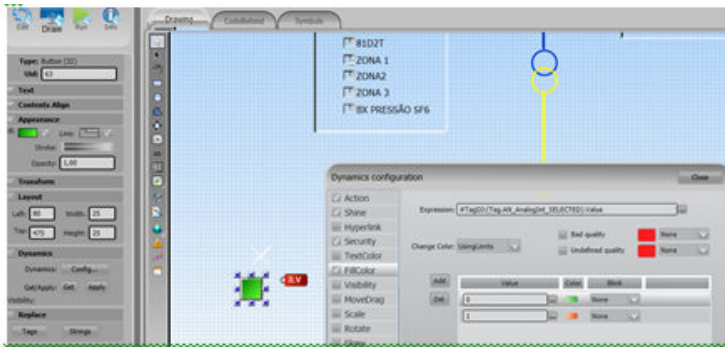
1. Faz que a variável global AN_DISJ_SELECTED seja o disjuntor selecionado:
   `Tag.AN_DISJ_SELECTED.Link=#TagDJ:(Tag.SEL_LINHA_01.CB.DJ).GetName()`
2. Chama a janela de comando:
   `Display.AN_Command.Open()`



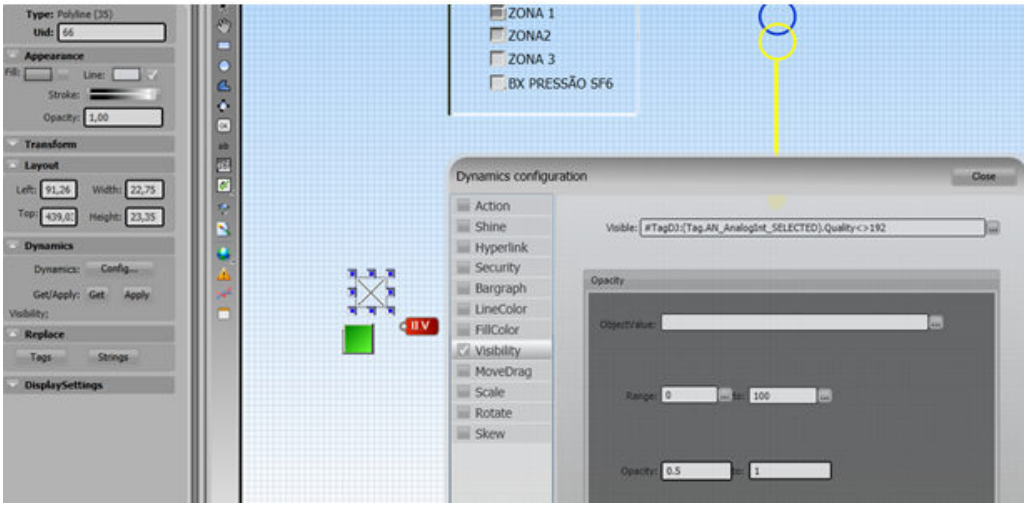
- O mouse sobre o objeto aumenta seu tamanho de 20% e sua opacidade é 0%.



- Quando a variável é "0", a cor é verde e quando é "1" a cor é vermelha:



- O "X" são duas linhas agrupadas e possuem uma dinâmica que é só aparecer quando a qualidade do tag é diferente de 192 (192 = qualidade OK).

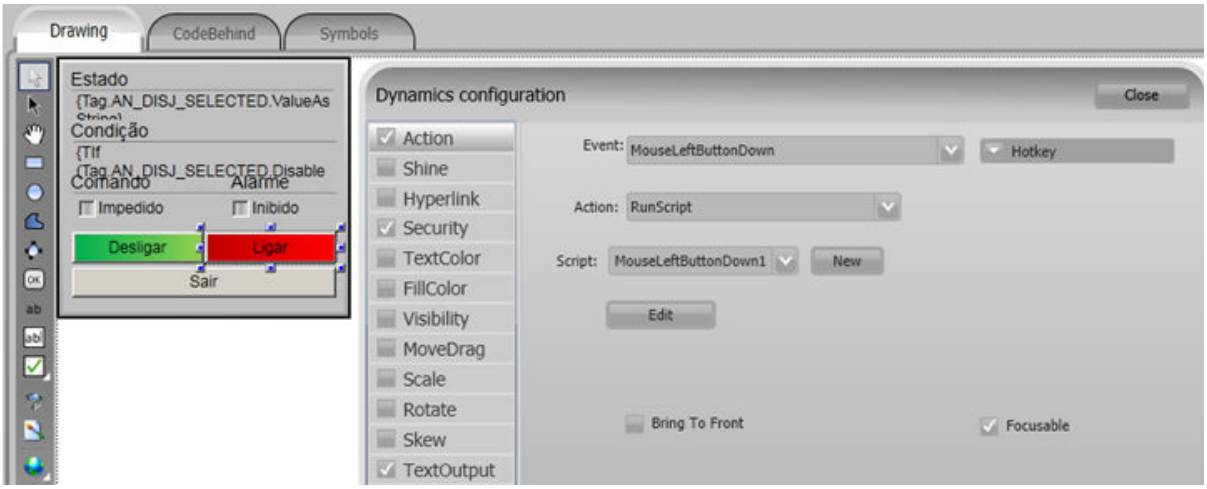


Janela de Comando (AN_Command)

Nesta janela existem três TextOutputs com os textos:

- Estado
- Condição
- Comando Alarme

Os botões de ligar e desligar possuem a dinâmica de Mouse clique (MouseLeftButtonDown) associadas aos Scripts MouseLeftButtonDown1 e MouseLeftButtonDown2, respectivamente, que enviam o comando de Ligar e Desligar para o campo:



Estas rotinas são apresentadas abaixo:

```
public void MouseLeftButtonDown1(object sender,
System.Windows.Input.MouseButtonEventArgs e)
{
// Ligar disjuntor / Seccionador
//
string txtAux;
string txtNome;
txtNome = @Tag.AN_DISJ_SELECTED.FirstDescription + " " +
@Tag.AN_DISJ_SELECTED.Description;
txtAux = @Tag.AN_DISJ_SELECTED.Link;
if (@Tag.AN_DISJ_SELECTED.Disable != 1)
{
      if (MessageBox.Show("Confirma comando LIGAR " + txtNome + " ?", "DESLIGAR",
MessageBoxButton.YesNo) == MessageBoxResult.Yes)
      {
      // Envia Msg para log de operação
      @Alarm.SystemEvents.AddCustomMessage("COMANDO FECHAR " + txtNome);
      //Envia Comando
      @Script.Class.AN_Tags.ToggleDigOutput(txtAux + "_L");
      }
}
else MessageBox.Show("CMD IMPEDIDO");
}
public void MouseLeftButtonDown2(object sender,
System.Windows.Input.MouseButtonEventArgs e)
```

```
{
// Desligar Disjuntor / Seccionador
string txtAux;
string txtNome;
txtNome = @Tag.AN_DISJ_SELECTED.FirstDescription + " " +
@Tag.AN_DISJ_SELECTED.Description;
txtAux = @Tag.AN_DISJ_SELECTED.Link;
if (@Tag.AN_DISJ_SELECTED.Disable != 1)
{
      if (MessageBox.Show("Confirma comando DESLIGAR " + txtNome + " ?", "DESLIGAR",
MessageBoxButton.YesNo) == MessageBoxResult.Yes)
      {
      // Envia Msg para log de operação
      @Alarm.SystemEvents.AddCustomMessage("COMANDO ABRIR " + txtNome);
      //Envia Comando
      @Script.Class.AN_Tags.ToggleDigOutput(txtAux + "_D");
      }
}
else
      MessageBox.Show("COMANDO IMPEDIDO");
}
```

Aplicação default Express

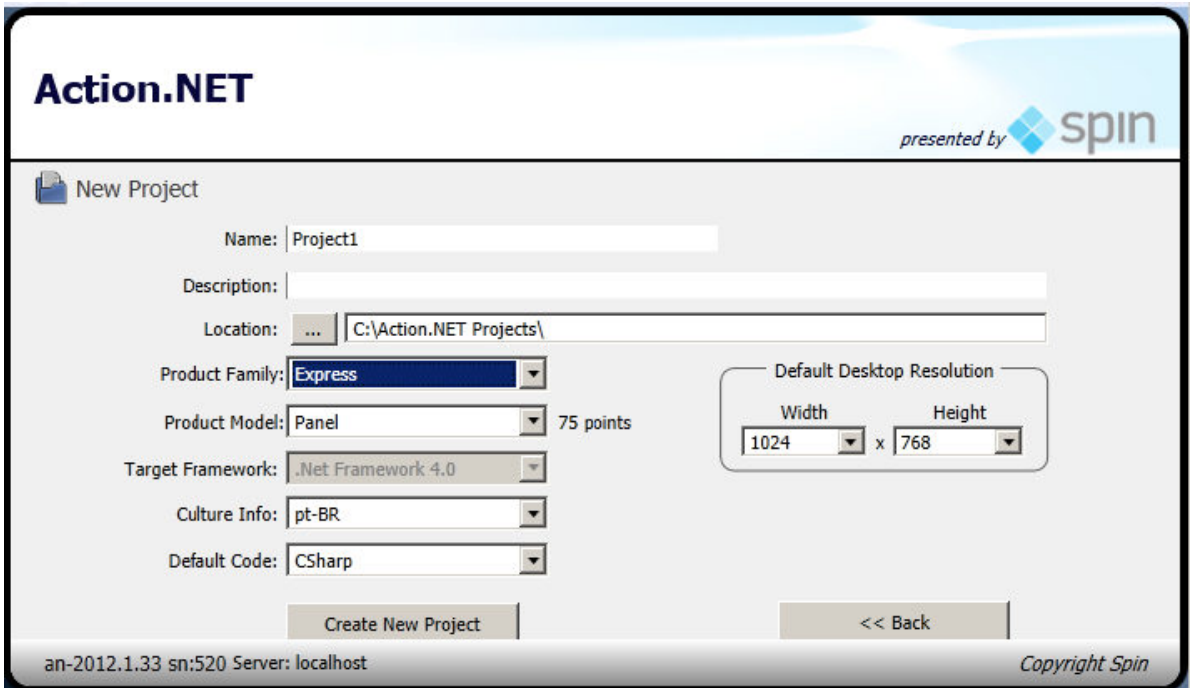
Introdução

A aplicação Express, até 75 pontos, pode ser obtida gratuitamente por integradores e instituições de ensino. Ela tem algumas limitações com relação a Enterprise que são:

- Máximo número de pontos:75;
- Número de diferentes protocolos simultâneos: 1;
- Número máximo de projetos em execução: 1;
- Número de engenheiros por projeto: 1;
- Array de tags: máximo uma dimensão;
- Nível de template: 1 (não é possível um template se referencias a outro template);
- Redundância: não suportada;

Objetivo

De forma semelhante à aplicação Enterprise, na versão Express, quando o usuário cria uma nova aplicação, como apresentado na figura abaixo, esta já vem com um conjunto de tags, telas, alarmes e outros pré-prontos.



Esta aplicação é designada "Aplicação Express-Default" e, a partir dela, o usuário pode, rapidamente, gerar uma nova aplicação com telas de alarmes, eventos, log de operação, tendência em tempo real, etc.

A seguir são apresentadas as informações sobre a "Aplicação Express-Default" e como usá-la no desenvolvimento de uma nova aplicação.

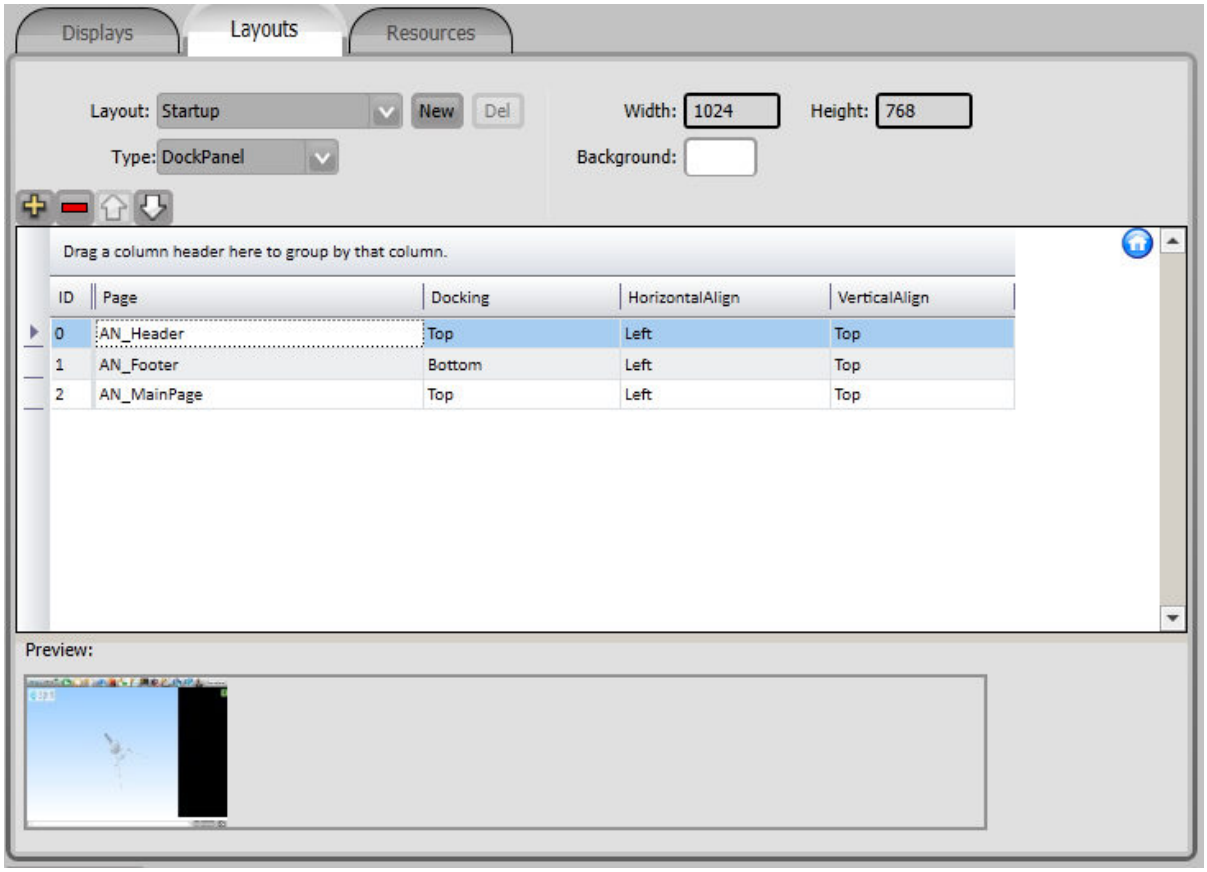
Deve-se observar que as informações apresentadas aqui deverão evoluir com o tempo, dado que a Spin, em função de solicitação de usuários, deverá sempre aprimorar a Aplicação Default, disponibilizando novas funcionalidades, Symbols, Figuras, etc.

Objetos da Aplicação Default Express

A Aplicação Express-Default já vem com um layout e algumas telas prontas, que o usuário poderá usar em sua aplicação.

Layout Disponível

Startup: tela com cabeçalho e rodapé, conforme a tela abaixo:



Este layout é usado para apresentar todas as telas da Aplicação Default.

Telas Disponibilizadas

Acima é apresentado o cabeçalho da aplicação default onde:

O símbolo quando selecionado, troca o idioma da aplicação default (inglês ←→/ português);

O símbolo retorna para a tela anterior;

O simbolo vai para a tela principal da aplicação default, que inicialmente é uma tela vazia com um compasso. O usuário pode alterar esta tela (AN_MainPage) para ser sua tela de abertura;

O simbolo abre a tela de alarmes correntes (AN_AlarmsSummary);

O símbolo abre a tela de eventos do dia (AN_EventsSummary);

O símbolo abre a tela de consulta a eventos históricos (AN_HistoricEventsSummary);

O simbolo abre a tela de tendência em tempo real e histórica (AN_TrendScreen);

O símbolo abre a tela de log de operação (AN_Operation_Log);

O símbolo cala o alarme sonoro;

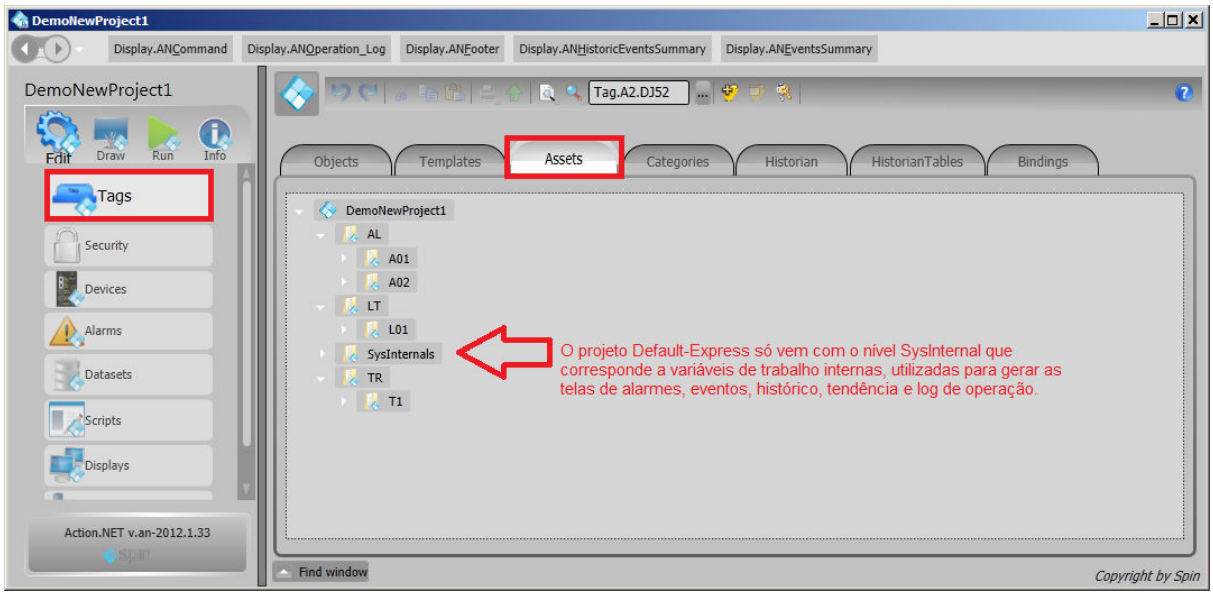
O símbolo abre a tela de informações do projeto (AN_About).

Quando a Aplicação Default Express é ativada, estas telas estão vazias e sem filtros, pois não existem variáveis do usuário nesta aplicação.

Tags da Aplicação Express-Default

Criação dos Níveis (Assets)

Todos os tags existentes na Aplicação Express-Default foram criados para gerar as telas já disponibilizadas, assim como fazer filtros destas telas que permitem selecionar coleções de tags associados à prioridade de alarme, categoria, nível, etc.



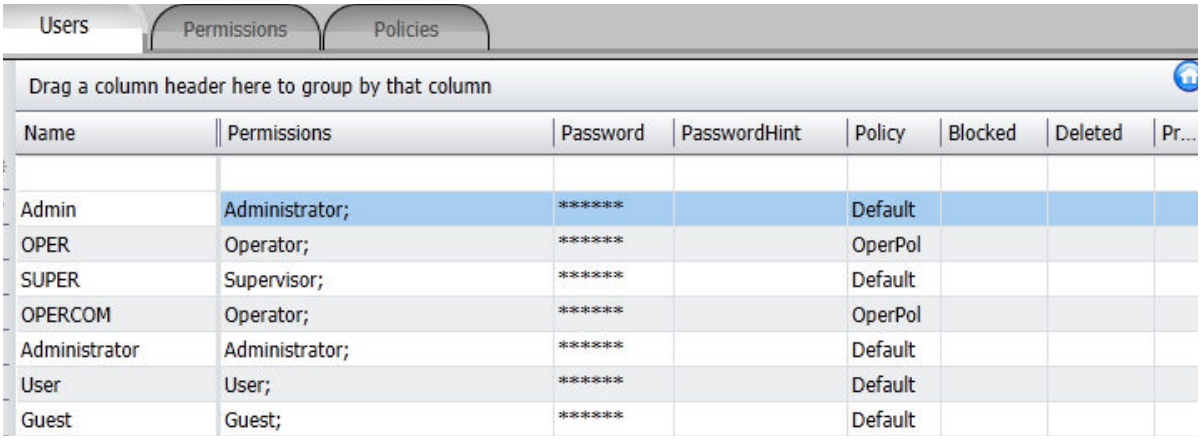
Estes tags são associados ao nível "*SysInternals*" que corresponde a variáveis internas da aplicação, não utilizáveis pelos usuários. Dessa forma, as variáveis que serão criadas pelo usuário e utilizadas nestas telas pré-existentes têm como restrições:

- Não podem ser do tipo SysInternals;
- Devem ser de nível (Asset) similar ao SysInternal ou superior;
- Devem ser do domínio do servidor (isto é, não podem ser variáveis associadas a uma estação de trabalho = Client);

Antes do usuário criar seus templates e tags, ele deve fazer a árvore da sua aplicação (Assets) que corresponderá aos níveis utilizados para filtrar variáveis. Em uma subestação, por exemplo, os níveis (1) são os tipos de bays e os níveis (2) são os bays propriamente ditos, como apresentado na figura acima, onde se tem uma subestação com uma linha, um transformador dois bays de alimentadores.

Security da Aplicação Express-Default

Na aplicação Default estão criados cinco usuários com as permissões conforme a figura abaixo:



Name	Permissions	Password	PasswordHint	Policy	Blocked	Deleted	Pr...
Admin	Administrator;	******		Default			
OPER	Operator;	******		OperPol			
SUPER	Supervisor;	******		Default			
OPERCOM	Operator;	******		OperPol			
Administrator	Administrator;	******		Default			
User	User;	******		Default			
Guest	Guest;	******		Default			

As senhas são:

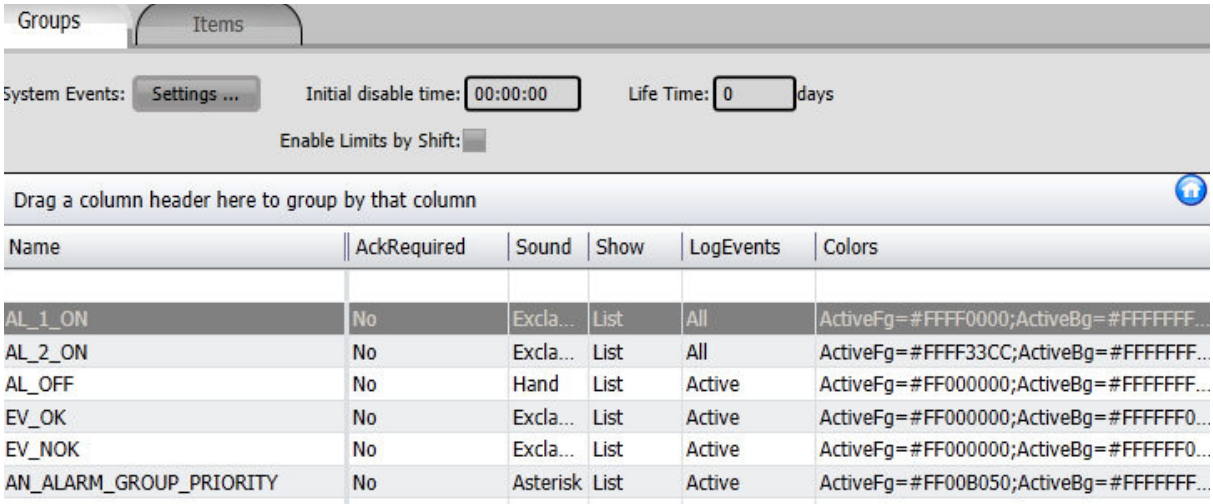
Gest → Administrator → sem senha

User → u Super → s

Operador → oper

Alarms da Aplicação Express-Default

Na Aplicação Express-Default já estão criados oito comportamentos de Alarmes que tem um procedimento padronizado com relação a sua inserção nas telas geradas automaticamente. A figura abaixo mostra estes alarmes:



Destes comportamentos, cinco são inseridos no *Sumário de Alarmes* da aplicação Default que corresponde a todos os pontos que estão em situação de alarme, atualmente, e três referem-se a eventos.

Eventos são todas as ocorrências que devem ser registradas em histórico e alarmes são ocorrências que devem ser corrigidas, pois se referem a situações de falha na aplicação.

Os alarmes estão classificados em graves e não graves e seus comportamentos são:

- AL_1_ON: Ocorrência de alarme grave;
- AL-2_ON: Ocorrência de alarme não grave;
- AL_ANA1: Ocorrência de LoLo / HiHi de analógico;
- AL_ANA2: Ocorrência de Lo / Hi de analógico;

Na Aplicação Default os alarmes devem ter prioridade 1 ou 2 ou 3, pois estes estão programados para serem inseridos no sumário de alarmes.

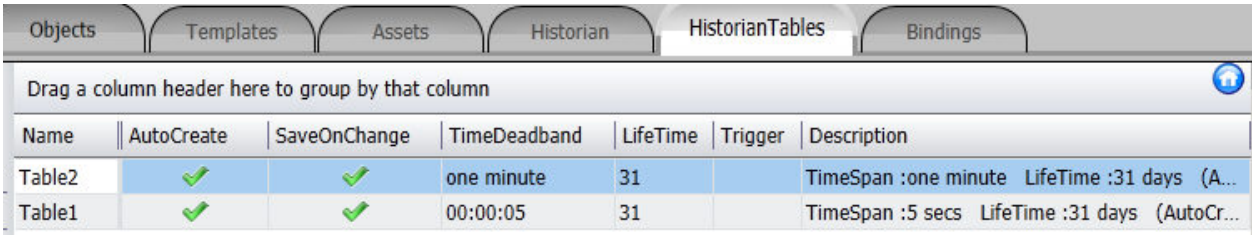
Todos os alarmes serão inseridos na lista de eventos também e, além destes, o *Sumário de Eventos* também terá a ocorrência de eventos que são de quatro tipos:

- EV_OK / EV_NOK: estes dois comportamentos são sugeridos para tags que têm um comportamento "mais normal" e um comportamento "menos normal", embora não seja um erro. A distinção será em cor de letra e cor de fundo.
- AL_OFF: Fim da ocorrência de um alarme seja ele grave ou não.
- NA_CHANGE: eventos que o valor do tag tem o mesmo significado (0,1,2, etc.).

Os eventos devem ter prioridade acima de 3.

Historian Tables

Na Aplicação Express-Default estão criadas duas tabelas: Table1 e Table2. A Table1 é atualizada a cada 5 segundos e a Table2 a cada minuto, conforme mostrado na figura abaixo:


Objects
Templates
Assets
Historian
HistorianTables
Bindings
Drag a column header here to group by that column
Name
AutoCreate
SaveOnChange
TimeDeadband
LifeTime
Trigger
Description
Table2
one minute
31
TimeSpan :one minute LifeTime :31 days (A...
Table1
00:00:05
31
TimeSpan :5 secs LifeTime :31 days (AutoCr...

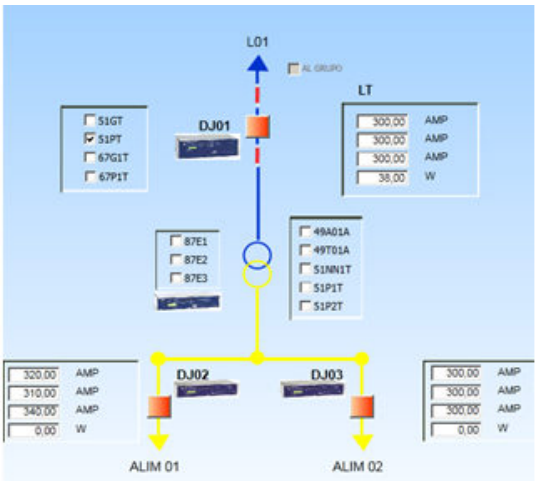
1.

Criação de Aplicação a partir Default-Express

Introdução

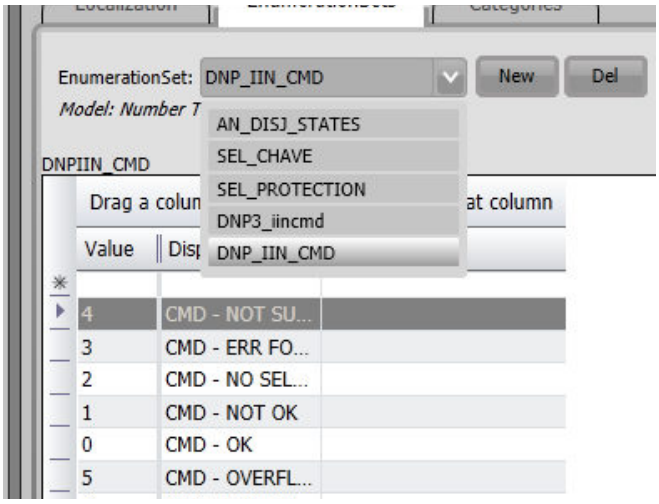
Para testar a aplicação Express-Default foi criado uma aplicação utilizando o protocolo DNP3 com 20 pontos de entrada e saída. O usuário pode utilizar esta aplicação para melhor entender o uso do software.

A aplicação exemplo possui um vão de linha, um vão de transformador e dois alimentadores, conforme mostrado no unifilar abaixo.Para todos os vão existe um relé associado.



Criação dos Dicionários

A aplicação Express-exemplo possui cinco dicionários, conforme mostra a figura abaixo:



Os dicionários tem cada uma sua finalidade:

1. Inglês: traduz palavras da aplicação para o inglês / português.
2. AN_DISJ_STATES: termos associados às posições 0 e 1 do disjuntor.
3. DNP_IIN_CMD: Termos associados aos internal indicators (ver manual do driver DNP 3.0) do DNP 3.0.
4. SEL_CHAVE: Termos associados as posições 00, 01, 10 e 11 das seccionadoras;
5. SEL_PROTECTION: Termos associados aos valores 0 e 1 dos tags das proteções.

Criação dos Templates

Foram criados três templates, um para o relé de proteção da linha (SEL_311L), um para o relé de proteção do transformador (SEL_387A) e um para os alimentadores (SEL_351A).

Como a aplicação Express não suporta template de template, todos os pontos de cada tipo estão definidos no seu template.

Na descrição dos tags é possível identificar seu conteúdo pelos comentários colocados e, como esta é uma aplicação exemplo, nem todos os tags de cada tipo (LT, TR e AL) foram endereçados.

Abaixo, a título de exemplo, é apresentado o template do vão de linha, as regras de nomeação utilizadas são:

- Prefixo <ANA>: variável analógica;
- Prefixo <DJ52>: disjuntor; Os comandos do disjuntor serão DJ52_L (ligar) e DJ52_D (desligar);
- Prefixo <SCn>: seccionadora <n>;
- Os comandos da seccionadora serão SCn_L (ligar) e SCn_D (desligar);
- Prefixo P: evento associado à proteção;

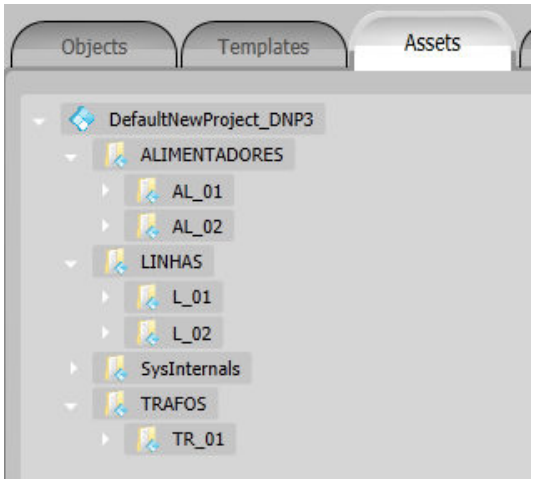
Criação de Níveis (Assets)

Os Assets (níveis) correspondem à organização dos agrupamentos de pontos da aplicação. Os tags tratados a partir da aplicação default devem ser diferentes de **SysInternals** que é o agrupamento dos tags usados nas rotinas internas da aplicação Default, para gerar as telas com os diversos relatórios disponibilizados automaticamente:

- Sumário de Alarmes;
- Sumário de Eventos;
- Eventos Históricos;
- Log de Operação;
- Tendência em Tempo Real;

Usando **NewLevel** definiu-se, conforme figura a seguir, três novos item de segundo nível, conforme mostra a figura abaixo:

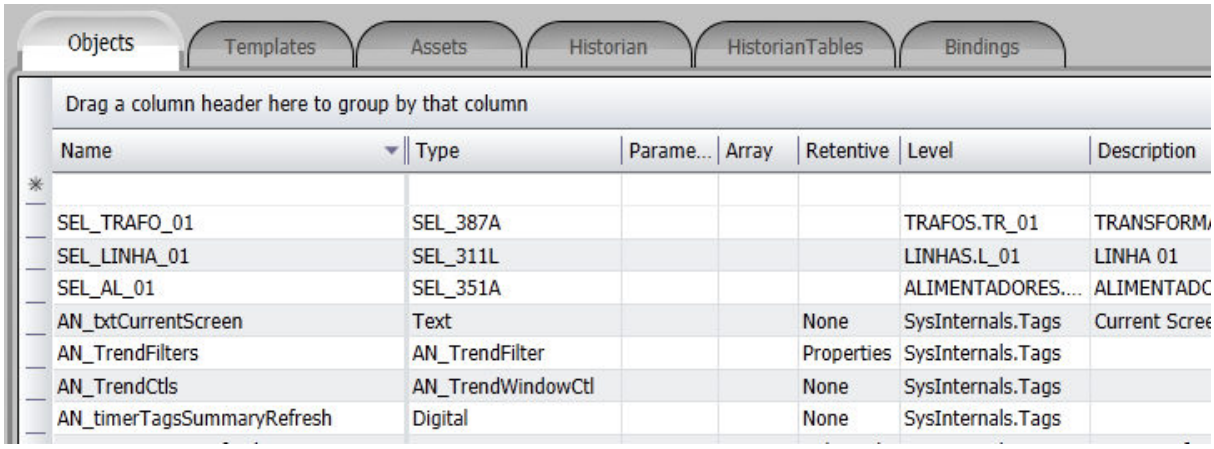
- LT: terá os bays de linha;
- TR: terá os bays de trafo;
- AL: terá os alimentadores.



Todos os tags criados deverão estar em um destes níveis.

Criação dos tags de Bays

Para criar os bays da subestação, vai-se para EDIT, TAGS e Objects e criam-se os bays associando-se seus tipos (Type) aos respectivos templates, conforme a figura abaixo.



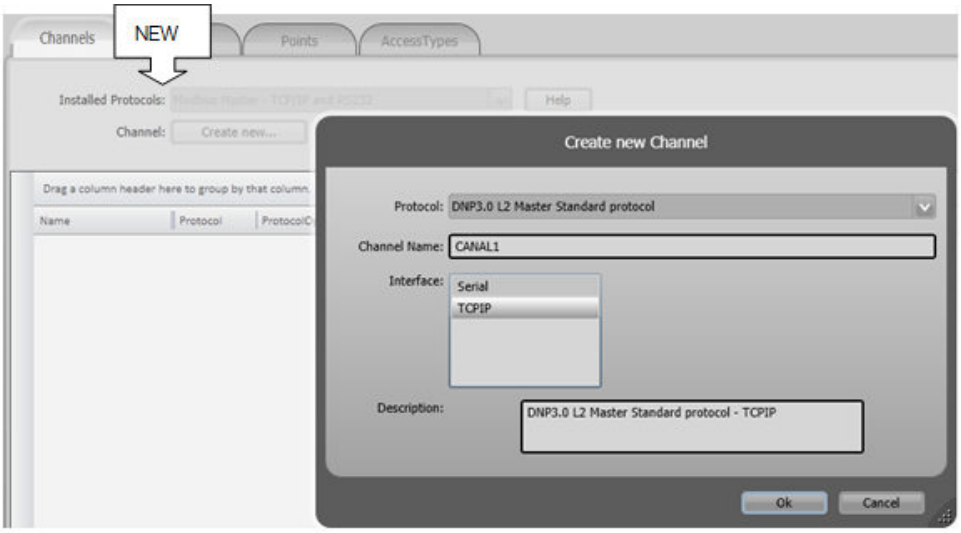
Observar que estes tags devem ser do tipo Server (estão no servidor) e não podem estar incluídos no nível *SysInternal*.

Criação do Device

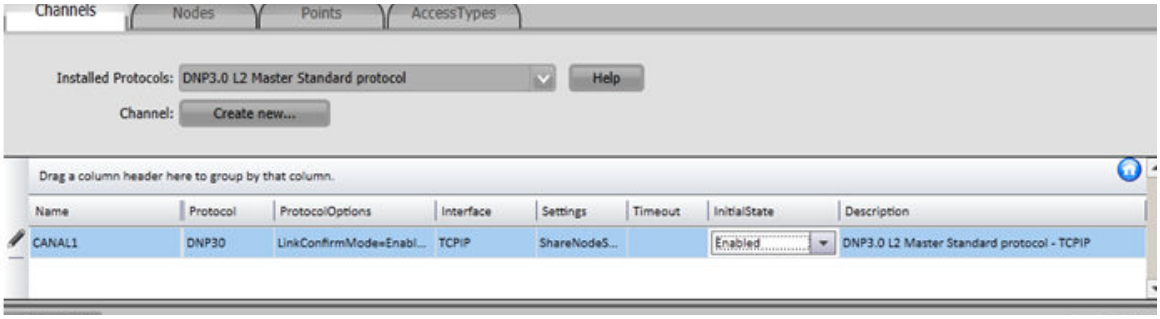
Nesta aplicação exemplo, considera-se que os relés estejam se comunicando com o SCADA através do protocolo DNP 3.0. Para cada protocolo existe um manual específico.

Criação de um Canal DNP3

Na Aba de canais (Channel) cria-se um novo canal chamado "CANAL1" do tipo DNP3 com comunicação TCP/IP, conforme figura abaixo:

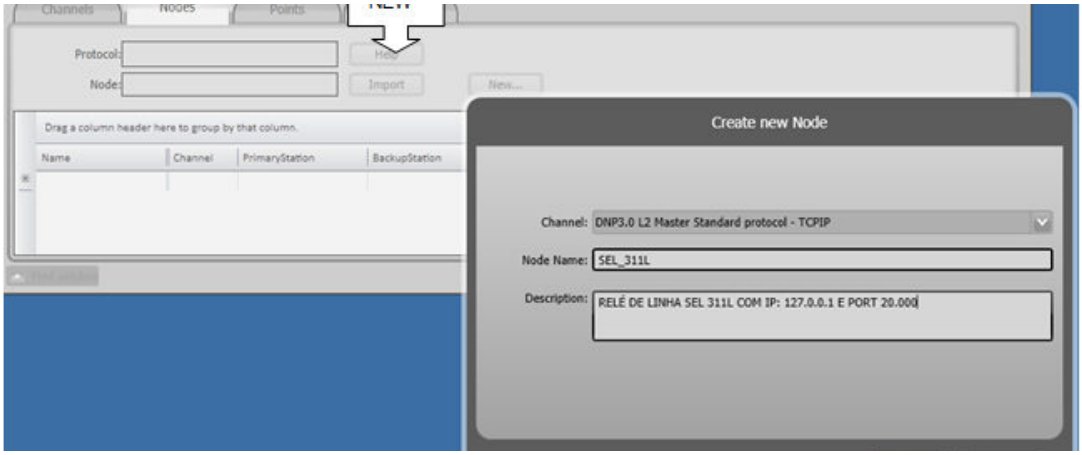


Neste canal se usa os atributos default, conforme abaixo:

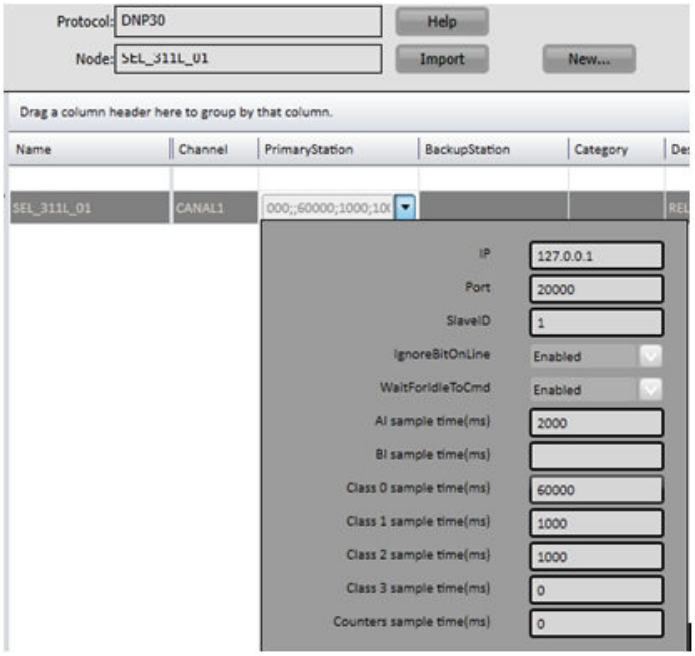


Criação do Nodo

O nodo corresponde ao IED (relé, CLP, UTR, etc.). Na ABA Node deve-se criar um novo nodo conforme a figura abaixo:

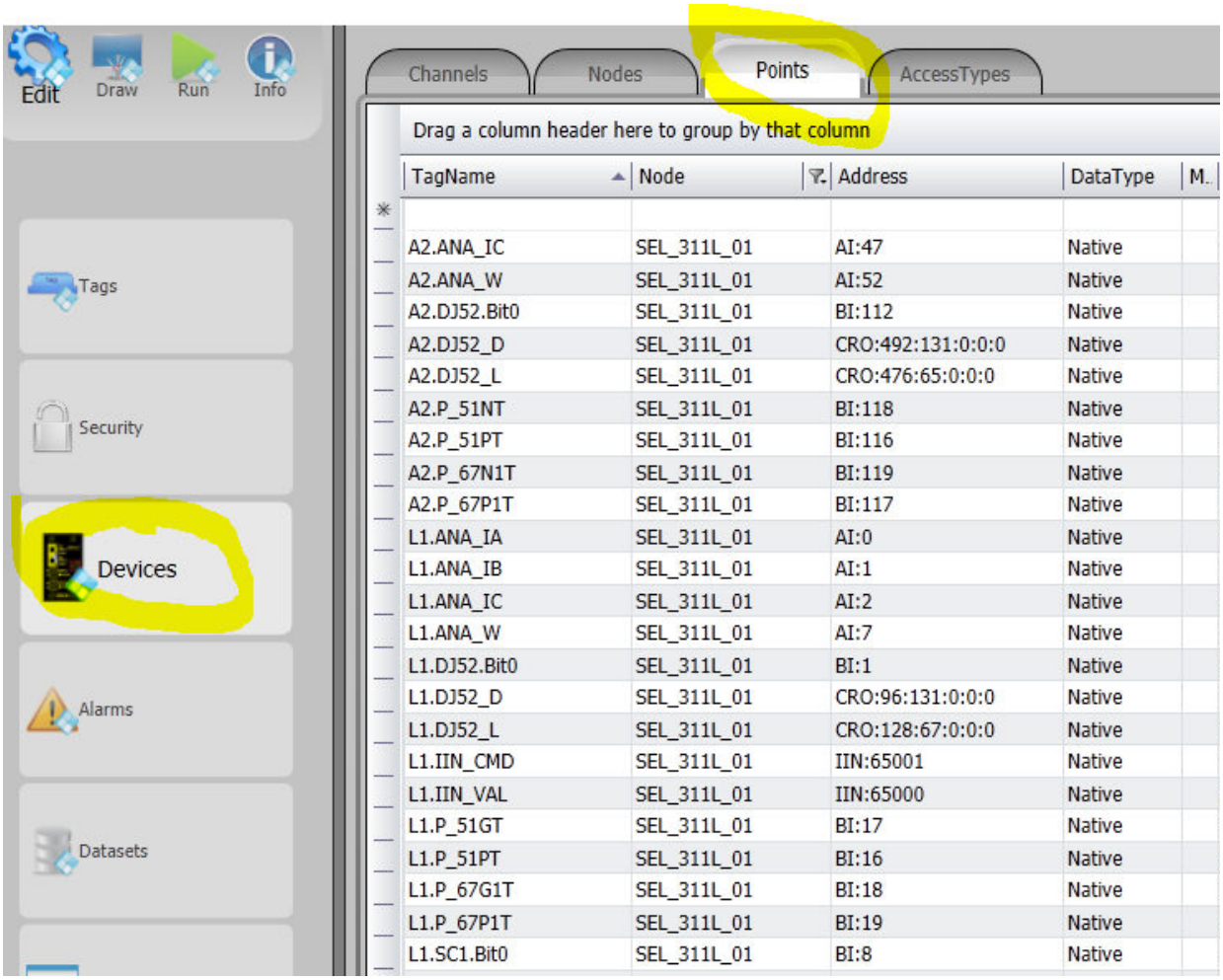


Após, deve-se acertar os atributos do node, conforme figura abaixo:



Criação dos Pontos do Canal / Nodo

Como esta é uma aplicação exemplo, ao invés de criar-se todos os pontos dos vãos (template), criou-se uns poucos pontos de cada um totalizando, para os quatro vãos, 53 pontos (o primeiro ponto é o zero).



TagName	Node	Address	DataType	M.
A2.ANA_IC	SEL_311L_01	AI:47	Native	
A2.ANA_W	SEL_311L_01	AI:52	Native	
A2.DJ52.Bit0	SEL_311L_01	BI:112	Native	
A2.DJ52_D	SEL_311L_01	CRO:492:131:0:0:0	Native	
A2.DJ52_L	SEL_311L_01	CRO:476:65:0:0:0	Native	
A2.P_51NT	SEL_311L_01	BI:118	Native	
A2.P_51PT	SEL_311L_01	BI:116	Native	
A2.P_67N1T	SEL_311L_01	BI:119	Native	
A2.P_67P1T	SEL_311L_01	BI:117	Native	
L1.ANA_IA	SEL_311L_01	AI:0	Native	
L1.ANA_IB	SEL_311L_01	AI:1	Native	
L1.ANA_IC	SEL_311L_01	AI:2	Native	
L1.ANA_W	SEL_311L_01	AI:7	Native	
L1.DJ52.Bit0	SEL_311L_01	BI:1	Native	
L1.DJ52_D	SEL_311L_01	CRO:96:131:0:0:0	Native	
L1.DJ52_L	SEL_311L_01	CRO:128:67:0:0:0	Native	
L1.IIN_CMD	SEL_311L_01	IIN:65001	Native	
L1.IIN_VAL	SEL_311L_01	IIN:65000	Native	
L1.P_51GT	SEL_311L_01	BI:17	Native	
L1.P_51PT	SEL_311L_01	BI:16	Native	
L1.P_67G1T	SEL_311L_01	BI:18	Native	
L1.P_67P1T	SEL_311L_01	BI:19	Native	
L1.SC1.Bit0	SEL_311L_01	BI:8	Native	

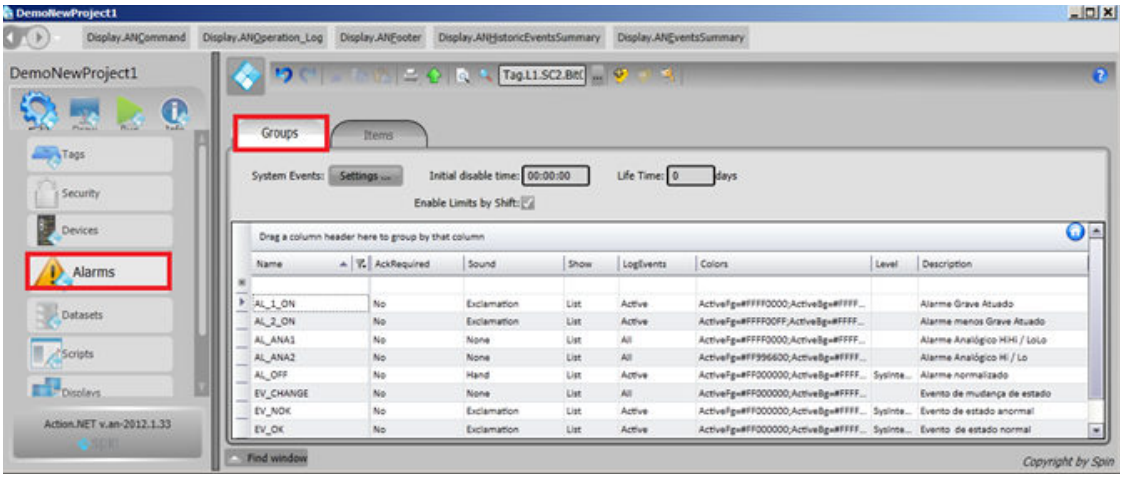
No manual do protocolo DNP3 existem as informações sobre como preencher cada coluna de um ponto. Os pontos BI são de entrada digital (Binary Input), os pontos AI são entradas analógicas, os pontos CRO são saídas digitais (Control Relay Output) e os pontos IIN são entradas do tipo Internal Indication do DNP3.

Observações:

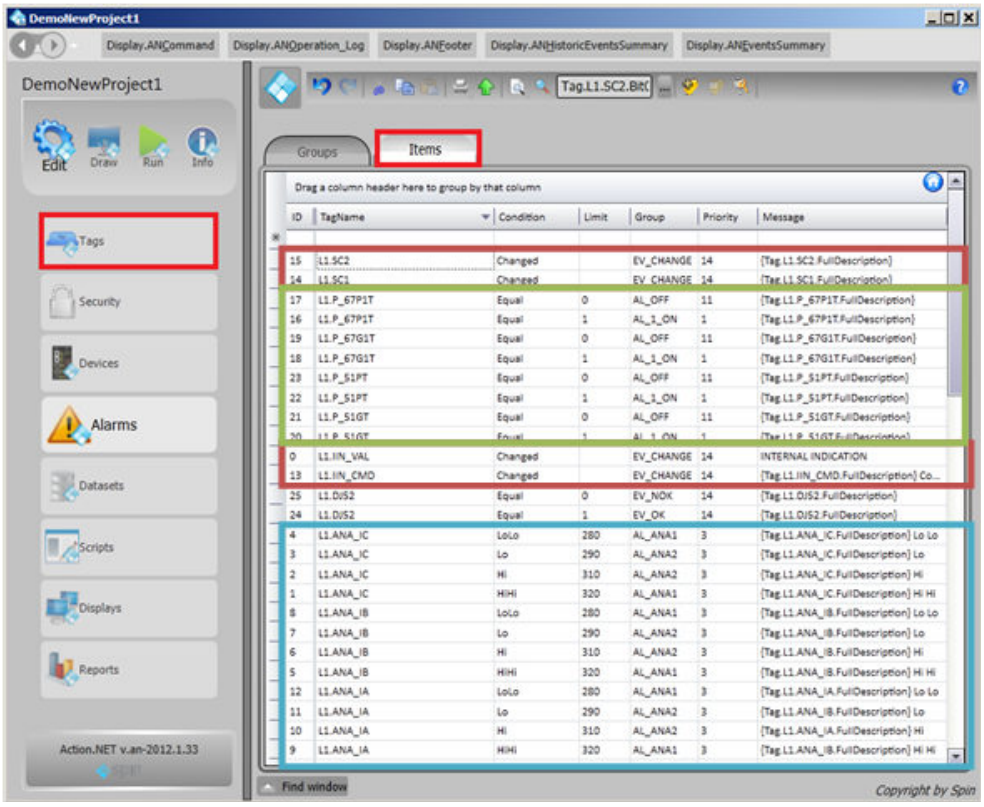
1. Para criar um ponto duplo como no caso das seccionadoras, definiu-se a seccionadora como uma variável AnalogInt, e criou-se dois pontos digitais que correspondem ao Bit0 e Bit1 da seccionadora (linhas selecionadas em azul – ID = 3/4);
2. As variáveis do tipo IIN (duas primeiras linhas da figura acima – ID = 19/20) têm endereço fixo, conforme apresentado no manual do DNP3;
3. Para executar uma saída deve-se chamar a rotina (Class) AN_TAG com método de inverter o valor da variável de saída:
4. *@Script.Class.AN_Tags.ToggleDigOutput(<tag de saída>)*;
5. Sempre que enviar um comando, é interessante colocar uma mensagem no Sumário de Eventos registrando este comando. Para tal execute a ação:
6. *@Alarm.SystemEvents.AddCustomMessage("Mensagem" + txtNome)* onde txtNome o string com a variável comandada.

Criação dos Alarmes e Eventos

A criação de alarmes, a partir da Aplicação Express-Default, exige que se crie na tabela de Itens de Alarmes uma entrada (Linha) para cada comportamento que deve ser sinalizado. A tabela abaixo mostra o número de entradas que serão criadas em função da situação:

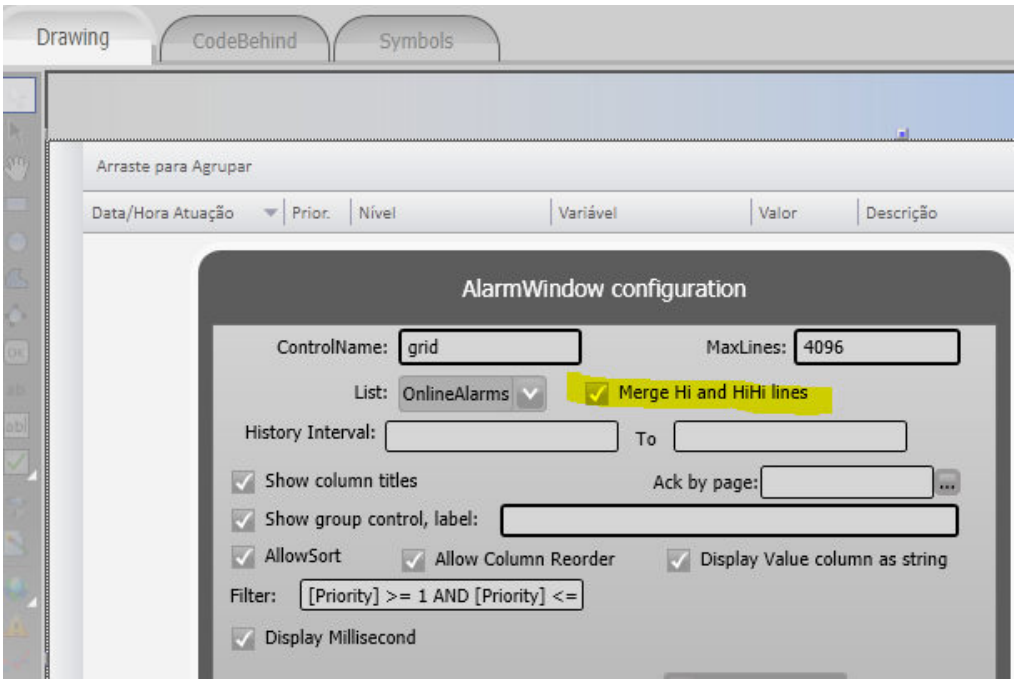


A tabela abaixo mostra uma lista de alarmes criada a partir dos Tags do Bay de linha usado como exemplo:



Observações:

1. As entradas circunscritas em azul referem-se a alarmes de Tags analógicos com dois limites superiores e 2 inferiores.
2. Quando um dos alarmes está atuado, os Tags são mantidos no Sumário de Alarmes e no de Eventos. Quando os Tags são normalizados, eles saem do Sumário de Alarmes e, no Sumário de Eventos é colocado o horário de sua normalização.
3. No ActionNET, quando da definição do Sumário de Alarmes, é selecionado o parâmetro "Merge Hi e HIHI" que mantém na lista o alarme de mais importância quando dois alarmes deste tipo estão atuados, isto é, se HIHI está atuado, a linha de HI é inibida e o mesmo ocorre com LO e LOLO.



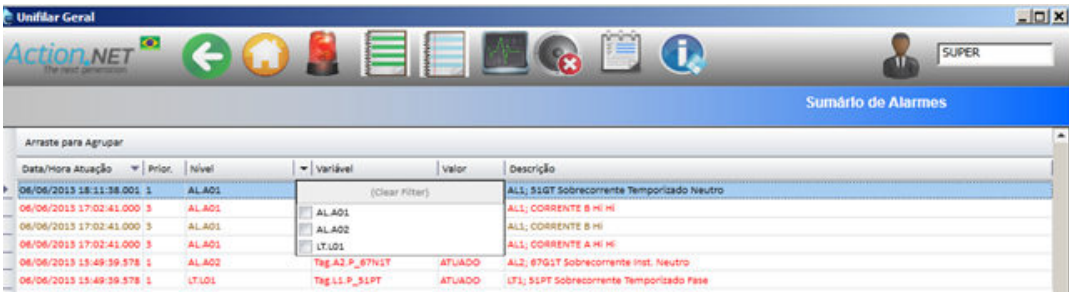
4. As entradas circunscritas em vermelho referem-se a eventos que são armazenados no histórico sempre que ocorre uma mudança de valor. Estes eventos são usados em manobras de equipamentos tais como seccionadora, tap de transformador, etc.
5. As entradas circunscritas em verde são alarmes atuados / não atuados, respectivamente.

A partir destas duas tabelas são geradas quatro telas:

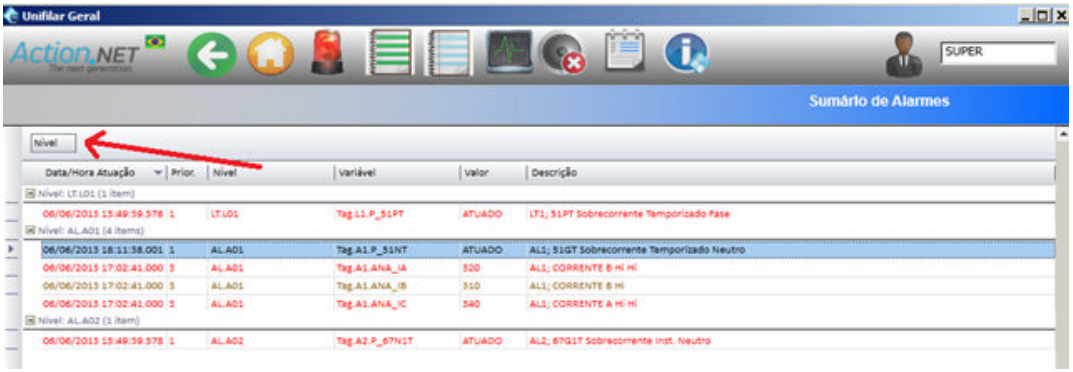
- Sumário de Alarmes;
- Sumário de Eventos;
- Eventos Históricos;
- Log de operação.

Em tempo de execução, o Sumário de Alarmes apresenta os alarmes que estão atuados no momento, como mostra a figura abaixo:

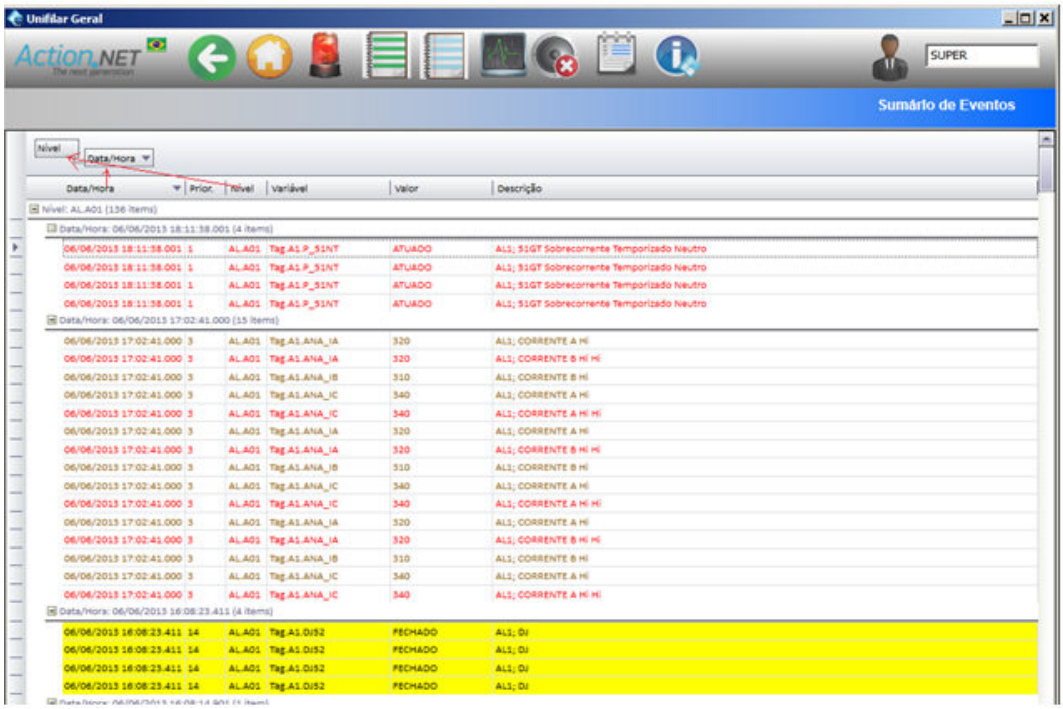
Sobre qualquer coluna pode ser aplicado um filtro para selecionar, por exemplo, apenas alarmes de um ou mais vãos:



Qualquer coluna pode ser arrastada para a linha superior, classificando os alarmes por esta coluna:



O Sumário de Eventos tem as mesmas colunas do sumário de alarmes apresentando em ordem cronológica todos os eventos e alarmes do dia, conforme mostrado abaixo. De forma semelhante aos alarmes, é possível arrastar uma coluna para a linha de cima, permitindo agrupar os eventos pela informação desta coluna. Se mais de uma coluna é arrastada, cria-se uma um agrupamento hierarquizado como apresentado abaixo.



Os Eventos Históricos mostram os eventos dentro do intervalo de dia e hora especificado em início e fim.

O log de Operação mostra todas as ações do operador;

Tela Gráfica de Tendências e Medidas históricas

O gráfico de tendência permite verificar tendência em tempo real e histórica de variáveis.

Atenção, as variáveis analógicas, obrigatoriamente, devem ser inseridas na tabela de Tags/Historian, para poderem ser vistas em tendências históricas.

No DataGrid o operador pode selecionar as variáveis, o intervalo de apresentação e se a tendência é de tempo real ou histórica. No segundo caso deve ser definida a data e hora inicial.

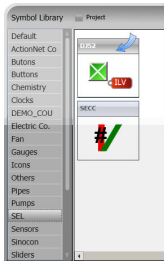
Criação das Telas da Aplicação

Conforme apresentado no capítulo referente à Aplicação Express-Default, já existem seis telas prontas, que poem ser chamadas pelos símbolos na barra de ferramentas da área de cabeçalho das telas. Veja símbolos de chamada das telas em "Telas Disponibilizadas" na página 255.

Janelas e Símbolos utilizados

Na aplicação Express-Default, além das telas de sumários e a tela principal, já existem outras janelas e simbolos:

- Janela de Comando (AN_Command)
- Janela de Login (LogOn)
- Biblioteca de símbolos prontos onde, para esta aplicação criou-se os dois símbolos apresentados abaixo:



Janela de Comandos (AN_Command)

Esta janela é usada para comandar disjuntores e seccionadoras. Ela apresenta o estado do equipamento (aberto, fechado e falha, no caso de equipamento com dois contatos), a condição de comando (habilitada / desabilitada) e botões para ligar / desligar equipamento e sair da janela.

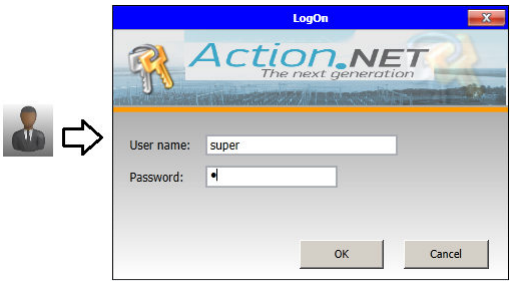
Quando um usuário clica em um objeto de visualização que será comandado por esta janela, o tag associado a este objeto de visualização é atribuído a variável tipo reference: **AN_Equip_Sel.**

As seguintes propriedades das variáveis definem seus estados:

- <AN_Equip_Sel>.Disable → 0 = Habilitado / 1= Desabilitado ou Impedido
- <AN_Equip_Sel>.Value → em digital simples → 0 = desligado / 1 = ligado
- <AN_Equip_Sel>.Value → em digital múltipla → 0 = falha / 1 = desligado / 2 = ligado / 3 = falha
- <AN_Equip_Sel>.AlarmDisable → 0 = alarme habilitado / 1 alarme desabilitado (não aparece nos sumários de eventos e alarmes.

Janela de Login (LogOn)

Quando é dado um clique do botão esquerdo do mouse na figura apresentada a esquerda, é aberta a janela de login de novo usuário.



Símbolo Disjuntor

É utilizado para apresentar o estado de um disjuntor. Este símbolo tem uma única variável que é <AN_DEFAULT_AnalogInt>. Quando o usuário insere este símbolo na tela, deve substituir esta variável por uma variável associada ao estado de um disjuntor, como, por exemplo, <L1.DJ52> que é o disjuntor da linha L1 na aplicação exemplo.

As dinâmicas deste símbolo são:

1. Quando o mouse passa sobre o controle, este aumenta 20% (Shine);
2. Disjuntor aberto→ cor verde, fechado → cor vermelha;
3. Quando <disjuntor>.disable = 1, é colocada uma bandeirola de Impedimento Linha Viva;
4. Quando feito um clique com botão esquerdo do mouse sobre o símbolo, o tag do disjuntor é associado a variável reference < Tag.AN_Equip_Sel.Link> e é chamada a janela de comando (AN_Command) que pressupõe existirem duas variáveis de saída com os nomes <disjuntor>_L e <disjuntor>_D que ligam e desligam o disjuntor, respectivamente.
5. Usuário tipo Gest não tem permissão para chamar a janela de comando.

Simbolo Seccionadora

É utilizado para apresentar o estado de uma seccionadora. Este símbolo tem uma única variável que é <AN_DEFAULT_AnalogInt>. Quando o usuário insere este símbolo na tela, deve substituir esta variável por uma variável associada ao estado de uma seccionadora posicionada verticalmente na tela como,

Por exemplo, *<L1_SC1>* que é uma das seccionadoras da linha L1 na aplicação exemplo.

As dinâmicas deste símbolo são:

1. Quando o mouse passa sobre o controle, este aumenta 20% (Shine);
2. Seccionadora aberta (1) Verde, fechada (2) → Vermelho, Inválida (0 ou 3) → símbolo de falha;
3. Quando feito um clique com botão esquerdo do mouse sobre o símbolo, o tag do disjuntor é associado a variável reference < Tag.AN_Equip_Sel.Link> e é chamada a janela de comando (AN_Command) que pressupõe existirem duas variáveis de saída com os nomes <seccionadora>_L e <seccionadora>_D que ligam e desligam o disjuntor, respectivamente.

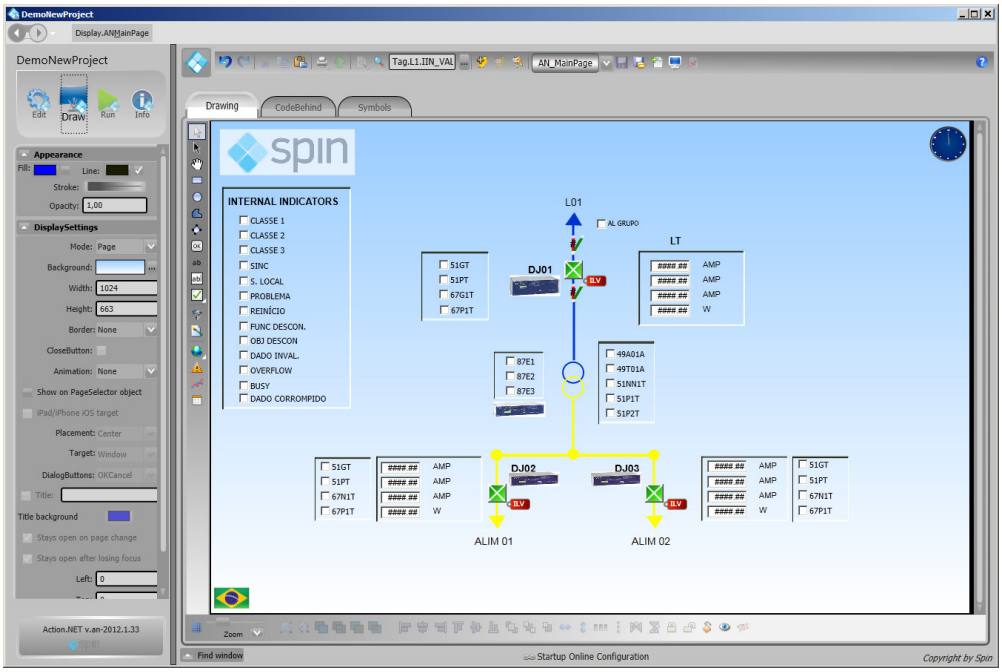
4. Usuário tipo Gest não tem permissão para chamar a janela de comando.

Tela Principal (AN_MainPage)

Na aplicação exemplo foi criada uma única tela que apresenta o unifilar da subestação com estado dos disjuntores dos bays de alimentador e linha e das duas seccionadoras da linha.

As proteções atuadas estão sendo sinalizadas através de CheckBox do tipo "output only". Existe um alarme de grupo que identifica se existe alguma proteção atuada na linha e os Internal Indicators do DNP3 também estão sendo sinalizados através de CheckBox.

Para a linha e os alimentadores são mostradas as correntes e a potência.



Utilizando ferramentas de diagnose

Introdução

O software Action.NET possui duas ferramentas de diagnóstico que podem ser utilizadas para verificar o comportamento de uma aplicação em desenvolvimento: Janela WATCH e janela TRACE.

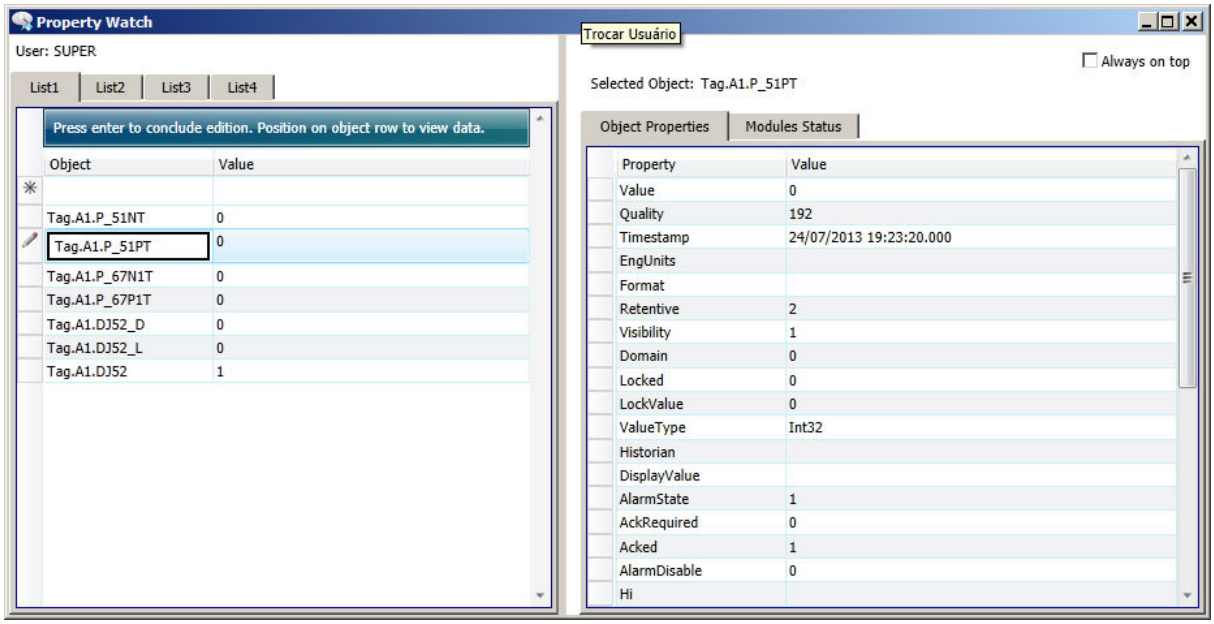
Janela Watch

Esta ferramenta é uma janela dividida em dois quadros, conforme mostra a figura abaixo, cujos principais objetivos são:

- Simular off-line o valor de objetos e de suas propriedades;
- Acompanhar, on-line, o estado dos objetos e suas propriedades:

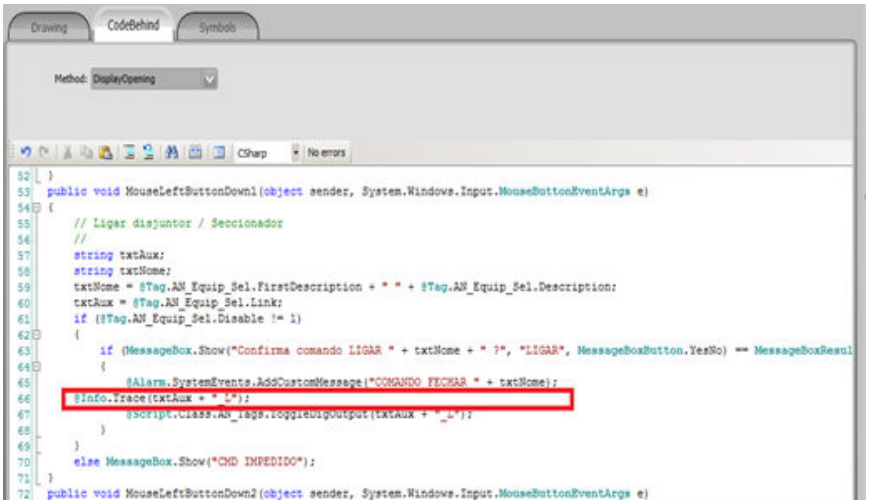
Na figura pode-se visualizar:

1. O quadro da esquerda, possui quatros abas (List1 a List4) onde o projetista da aplicação pode agrupar objetos de um mesmo tipo. Assim, por exemplo, na aba <List1> da figura abaixo se colocou tags de um mesmo alimentador (A1).
   - Se o projetista não estiver conectado aos IEDs e alterar o valor do tag, manualmente, serão executadas todas as ações da aplicação associadas a esta alteração.
   - Se o projetista estiver conectado a algum IED, on-line, o valor de cada tag cuja origem é o IED será o valor que está vindo do campo.
2. O quadro da direita, contém as propriedades do objeto selecionado no quadro da esquerda. No exemplo abaixo, propriedades do objeto <tag.S1.P_51PT>.
   - Se o projetista não estiver conectado aos IEDs, ele pode alterar o valor de uma propriedade, manualmente, e verificar se os tratamentos previstos para esta alteração ocorrerão.
   - Se o projetista estiver conectado a IEDs, os valores destas propriedades referem-se a condições associadas ao objeto. Assim, no exemplo, o objeto <tag.S1.P_51PT> possui qualidade = 192 (ponto OK), sua última leitura foi no dia 24 de julho às 19:23:20:000 (estampa de tempo), etc.



Janela Trace

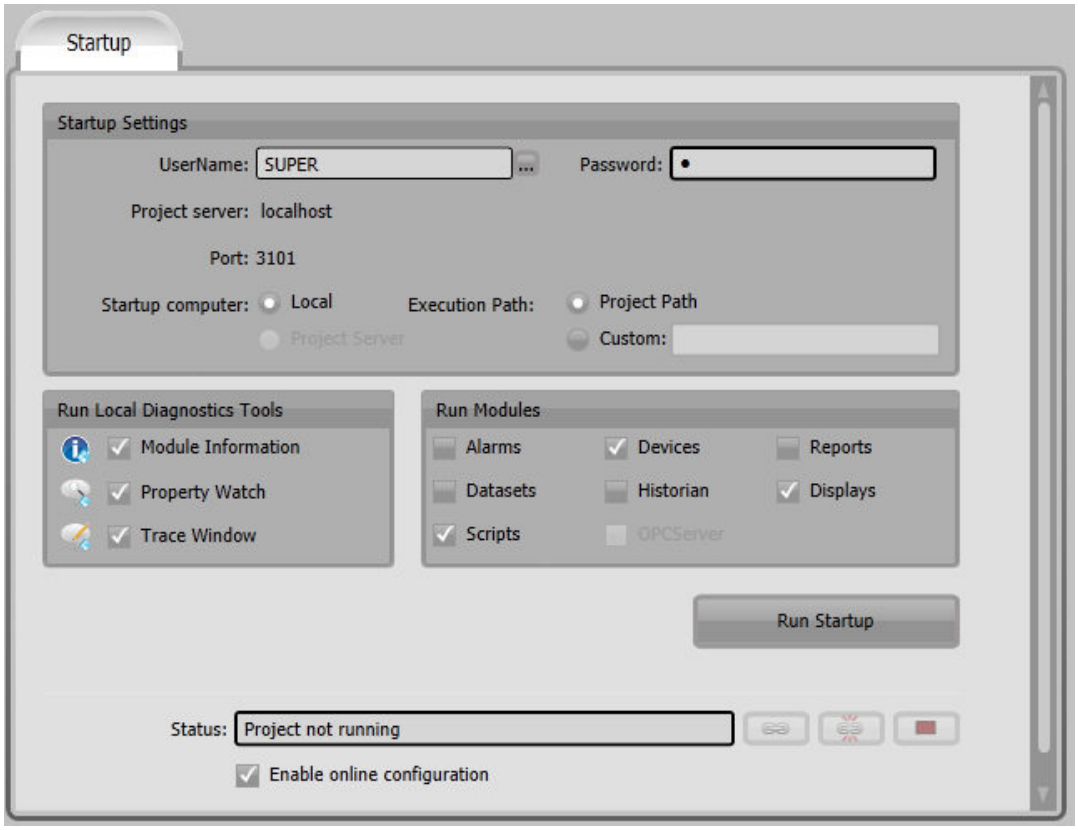
Esta ferramenta permite que se depure rotinas associadas aos diversos módulos do programa Action₀NET. Assim, por exemplo, se foram desenvolvidos vários códigos de Scripts associados a uma janela de comando, pode-se definir que o módulo de Display (responsável por apresentar telas da aplicação) seja rastreado e, dentro do código dos Scripts, pode-se colocar comandos que incluem informações na janela do trace, como o comando abaixo, sublinhado em vermelho: <@Info.Trace(txtAux + "_L");> que irá colocar na janela de Trace o valor da variável Txtaux concatenada com o string "_D".



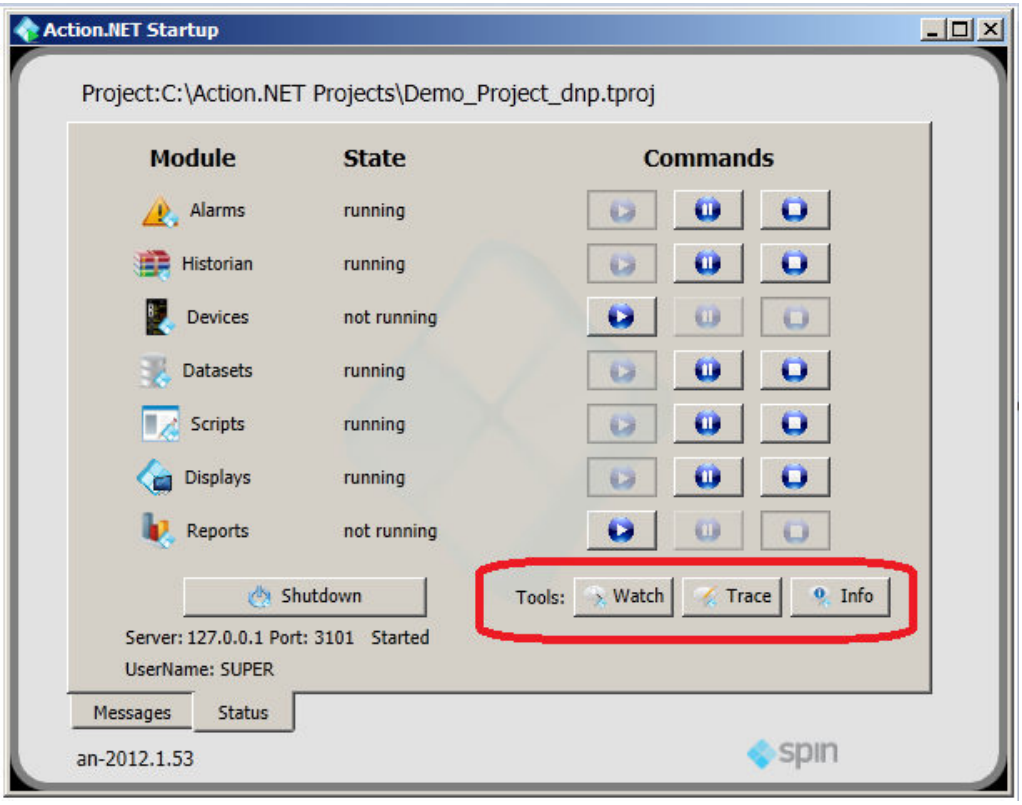
Ativando as Ferramentas de Diagnóstico

Quando o usuário vai ativar o run-time da aplicação, é possível definir quais ferramentas de diagnóstico serão ativadas junto com a aplicação (ver seleção em vermelho), assim como quais módulos da aplicação serão ativados (ver seleção em verde):

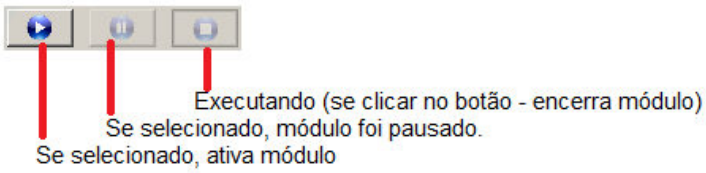
Uma vez ativada a aplicação, é apresentada a janela abaixo, com o estado de todos os módulos da



aplicação:



Na figura do exemplo acima, todos os módulos estão ativados, a menos do Devices e do Reports



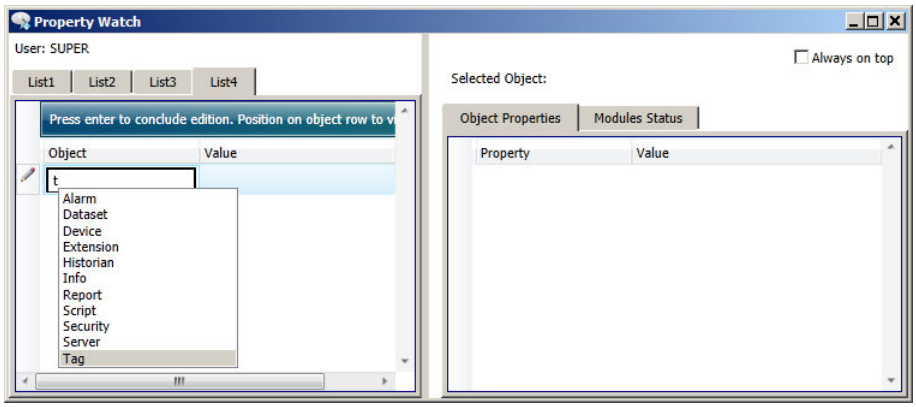
Assim, por exemplo, se o usuário faz uma alteração no módulo de Devices, que trata a comunicação com IEDs, basta parar o módulo e após reativá-lo.

Se ao ativar a aplicação o projetista não ativou o Watch ou Trace, basta pressionar os botões acima de Watch e Trace para iniciá-los.

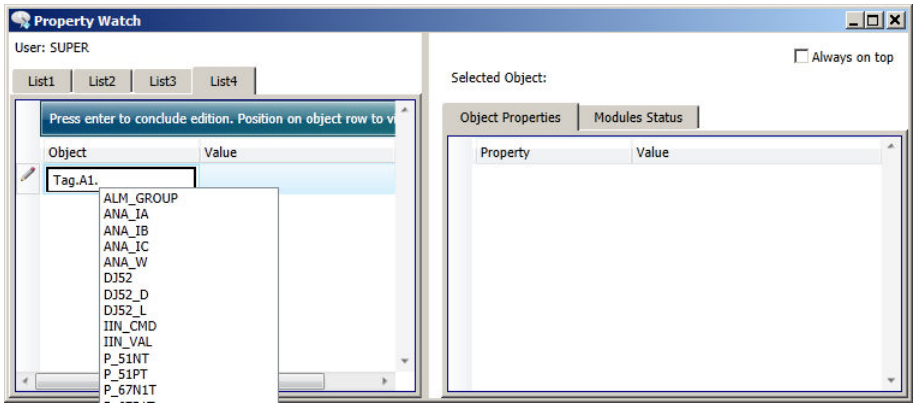
Usando o Watch

Quando ativado, este módulo vem com todos os campos vazios. Para inserir objetos em cada uma de suas abas, basta teclar a letra associada ao objeto (por exemplo, tag) que através da propriedade de "intellisense" (código inteligente) são mostrados os possíveis objetos associados àquela letra, como mostram as figuras abaixo:

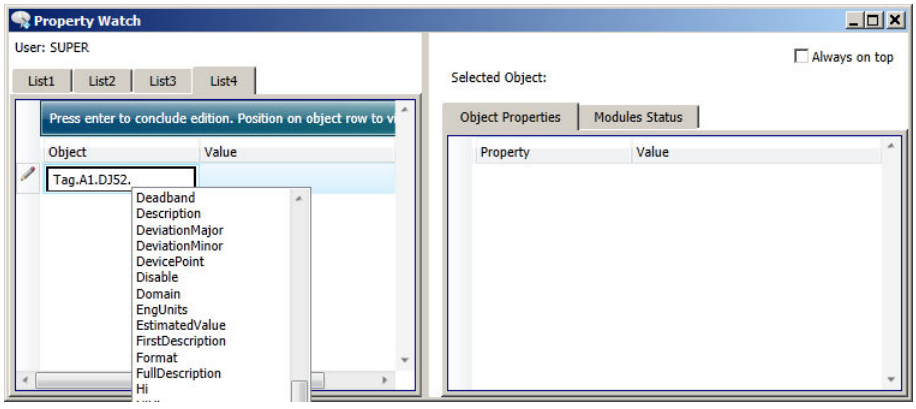
Ao digitar <t> são apresentados diversos objetos onde em destaque cinza <Tag> aparece o primeiro objeto que emparelha com a letra <t>.



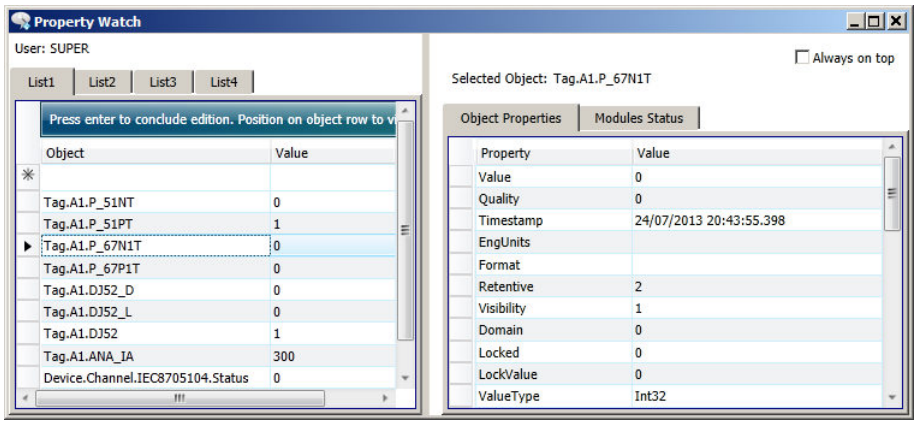
Foi inserido através de "Intellisense" o nome <tag.A1.> e ao ser inserido ponto após o A1 (alimentador 1) são apresentados todos os pontos deste alimentador.



Foi selecionado o ponto <Tag.A1.DJ52.> e ao ser inserido o ponto após o número 52 são apresentadas todas as propriedades associadas a este ponto. Dessa forma, é possível selecionar apenas as propriedades que se pretende depurar em um agrupamento de pontos.



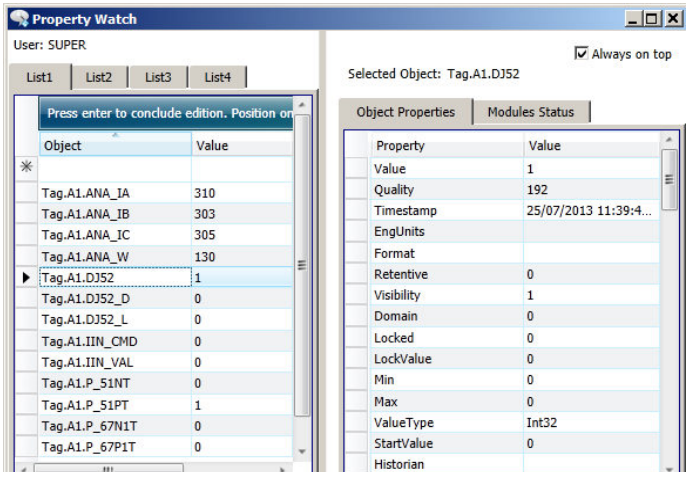
Concluindo, através do Watch adicionam-se vários objetos a uma aba da janela, os quais se pretende analisar o comportamento, como mostra a figura abaixo:



Quando a aplicação é encerrada, esta janela é mantida de forma que ao reiniciar a aplicação, todos os objetos colocados são mantidos, isto é, se em cada aba foram inseridos 20 objetos, os mesmos permanecerão quando a aplicação for reativada.

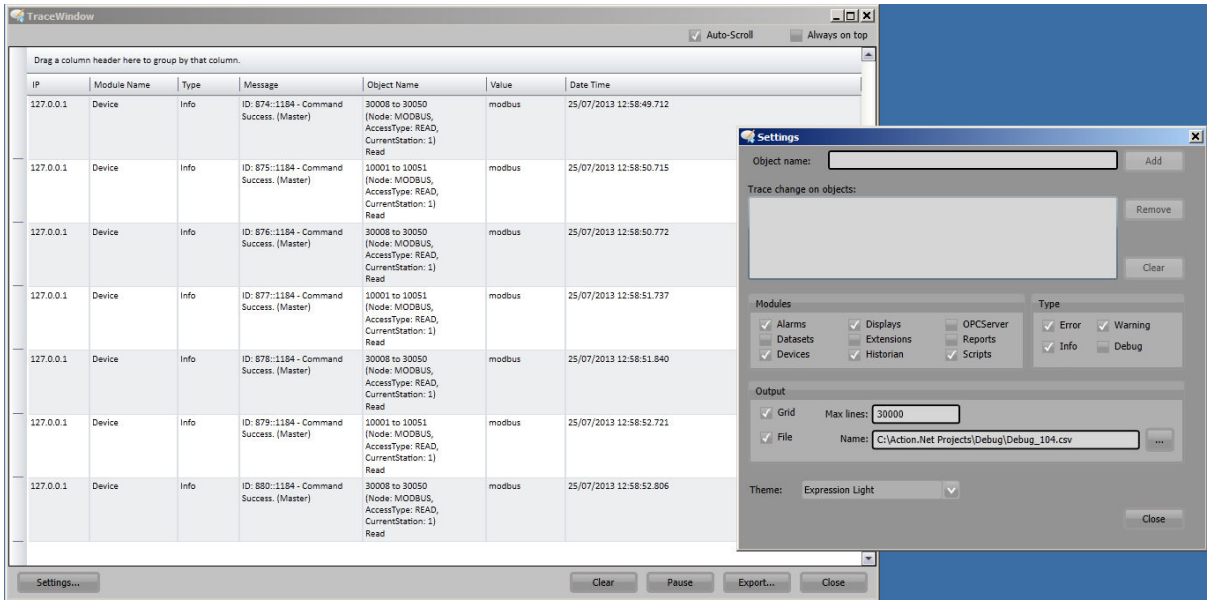
Uma forma mais rápida de se incluir pontos em uma aba é importá-los de uma planilha Excel, gerada como apresentada em capítulos anteriores. O exemplo apresentado na figura abaixo mostra uma planilha Excel gerada a partir do *template de um Alimentador*, onde adicionou-se o prefixo <Tag.A1> e o Header das colunas: *Object* / *Value*. Após gerar esta planilha, com todos os tags do Alimentador 1, basta copiar as colunas em cinza e cola-las em uma das abas da ferramenta Watch.

	Object	Value	Type	Array	Parameters
2	Tag.A1.IIN_VAL		AnalogInt		
3	Tag.A1.IIN_CMD		AnalogInt		Dictionary=DNP_IIN_CMD;
4	Tag.A1.P_51PT		Digital		Dictionary=SEL_PROTECTION
5	Tag.A1.P_51NT		Digital		Dictionary=SEL_PROTECTION
6	Tag.A1.P_67N1T		Digital		Dictionary=SEL_PROTECTION
7	Tag.A1.P_67P1T		Digital		Dictionary=SEL_PROTECTION
8	ALM_GROUP		Digital		Dictionary=SEL_PROTECTION
9	Tag.A1.ANA_IA		AnalogDouble		
10	Tag.A1.ANA_IB		AnalogDouble		
11	Tag.A1.ANA_IC		AnalogDouble		
12	Tag.A1.ANA_W		AnalogDouble		
13	Tag.A1.DJ52_D		Digital		
14	Tag.A1.DJ52_L		Digital		
15	Tag.A1.DJ52		AnalogInt		Dictionary=AN_DISJ_STATES;



Usando o Trace

O Trace, como dito acima, permite que se faça o rastreamento de módulos do Action.NET. Ao ser ativado, como mostrado na figura abaixo, é aberta uma janela para que o projetista da aplicação selecione as opções de depuração. O botão de <Settings> abre uma segunda janela com as possíveis opções.



Uma vez definidas as opções o usuário o projetista da aplicação fecha esta janela (Close) e ela ficará associada a este projeto sempre que o mesmo for ativado.

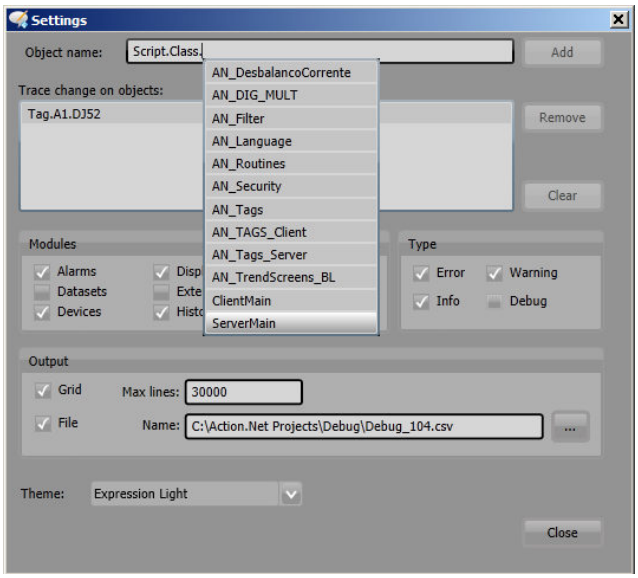
Os botões da janela de Trace são:

- Settings: abre a janela com opções de depuração;
- Clear: limpa todas as linhas desta janela;
- Pause: para de inserir linhas na planilha, em função da execução da aplicação;
- Export: Esporta os dados para uma planilha XML;
- Close: Fecha a janela de Trace.

Janela de Opções do Trace

Esta janela é subdividida em quatro secções:

1. Rastreamento de Objetos: permite selecionar objetos como, por exemplo, tags, Scripts, etc. que sempre que utilizados / alterados provocam a criação de uma nova linha na janela de Trace informando a modificação. Os objetos suportam "Intellisense" e os botões <Add>, <Remove> e <Clear> são utilizados para executar estas ações na lista.



2. Módulos que serão Rastreados: permite que o projetista selecione um ou mais módulos dentre os existentes no Action.NET. Além da seleção do módulo é possível selecionar os tipos de eventos associados ao módulo: Error, Warning, Info e Debug.
3. Outputs (saídas): define onde serão escritas as linhas de rastreamento, se na matriz de Trace e/ou em um arquivo no formato XML.
4. Tema da Janela: permite escolher a cor da janela dentre quatro possíveis.

Deploy - Geração automatizada de projetos

Introdução

Este assistente de importação pode ser utilizado para a geração automática de Tags, Alarm Items, Historian tables e a tabela de Points de um projeto. Para tal é necessário que se crie anteriormente Templates, Canais, Nodes e as Historian Tables que forem necessárias ao projeto.

Também devem ser criadas Categorias que indicarão como devem ser usados os templates e Points. O assistente ao ser executados faz a "explosão", i. é, popula com os registros adequados as tabelas mencionadas.

Cada cliente ou usuário poderá ter bases de dados com protótipos default apropriados a suas necessidades e sua cultura. Para criar estas bases o usuário deverá:

- Definir os Templates de todos os bays utilizando a regra de nomeação e alarmes própria da instalação do usuário;
- Definir o comportamento de geração de alarmes e eventos e escolher os Alarm Groups que serão usados no projeto;
- Definir as tabelas para o Historiador, isto é as Historian Tables.

Alarmes

A extensão de Geração automática de base de dados Deploy já define um conjunto de protótipos para os alarmes de grupo. Destes nomes deve-se escolher aqueles que atendem os requisitos dos clientes. Provavelmente, dos nomes abaixo, cada cliente utilizará cerca de uma dezena.

Os Alarmes de Grupo já disponíveis são listados abaixo. A identificação de cada AlarmGroup é relativamente intuitiva quanto a sua funcionalidade. Ao executar o Deploy uma primeira vez, estes grupos são criados e se houver dúvida quanto a funcionalidade, pode-se entender melhor olhando para o conjunto completo de campos do registro.

Utilizam **condition Change**

EVENTOS_BIP

EVENTOS_NOBIP

Utilizam **condition HI**

HI_ACK_BIP

HI_ACK_NOBIP

HI_NOACK_BIP

HI_NOACK_NOBIP

Utilizam **condition HIHI**

HIHI_ACK_BIP

HIHI_ACK_NOBIP

HIHI_NOACK_BIP

HIHI_NOACK_NOBIP

Utilizam **condition LO**

LO_ACK_BIP

LO_ACK_NOBIP

LO_NOACK_BIP

LO_NOACK_NOBIP

Utilizam **condition LOLO**

LOLO_ACK_BIP

LOLO_ACK_NOBIP

LOLO_NOACK_BIP

LOLO_NOACK_NOBIP

Utilizam **condition Equal ou NotEqual**

ACK_BIP

ACK_NOBIP

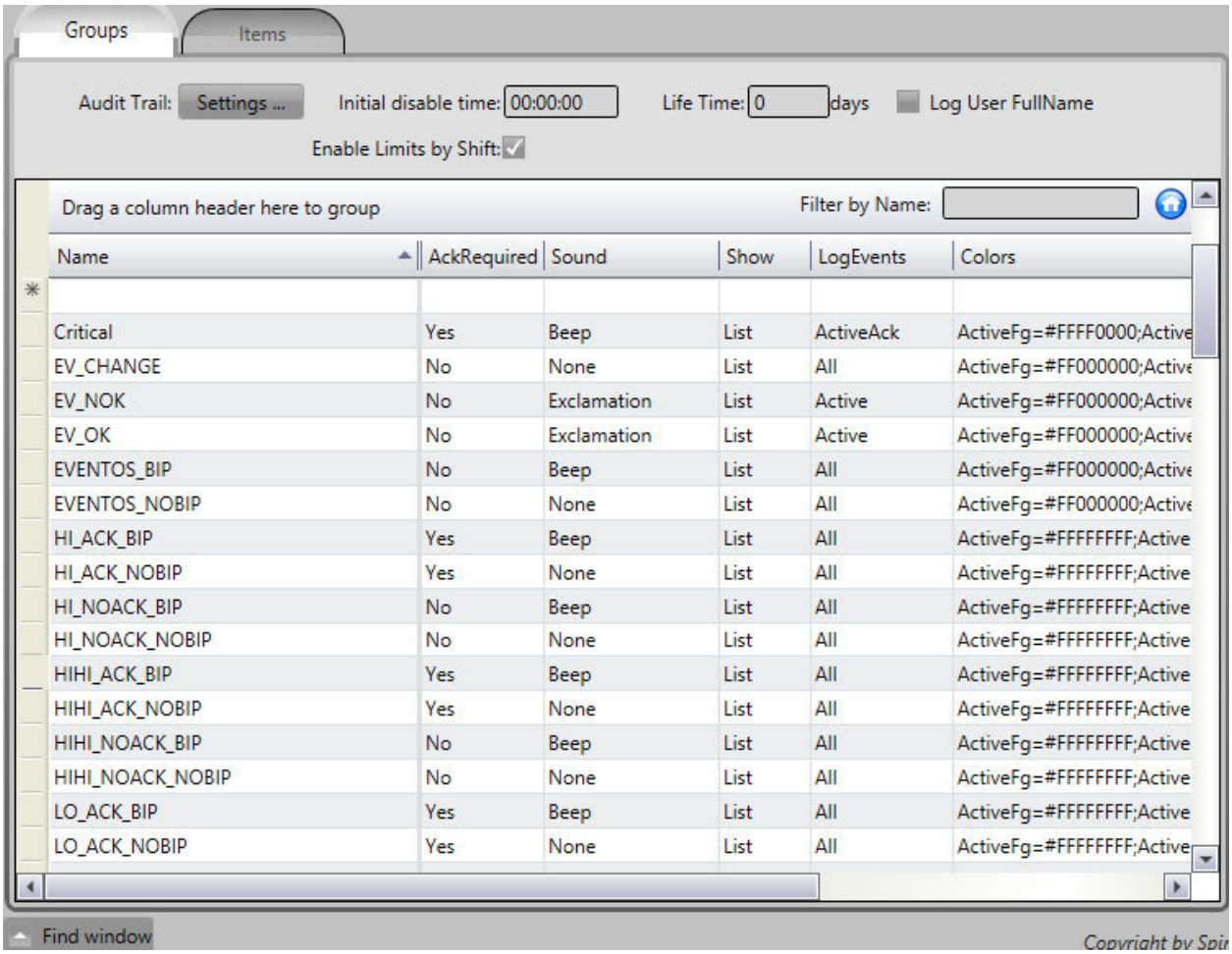
NOACK_BIP

NOACK_NOBIP

NOMSG_BIP

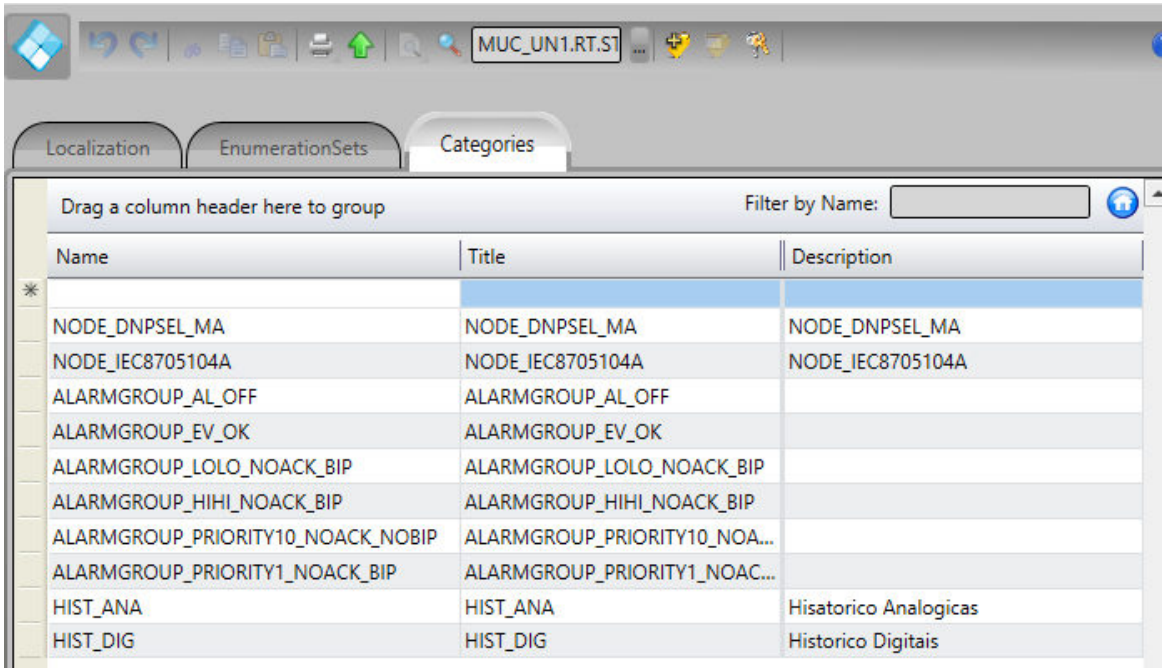
NOMSG_NOBIP

Observação: Para escolher Equal ou NotEqual, especifique junto com o estado (limite) em comments.



Para cada alarme de grupo realmente utilizado na Aplicação, deverá ser criada uma **categoria** correspondente com o prefixo: ALARMGROUP_ seguido do nome do alarme de grupo.

Exemplo: ALARMGROUP_NOMSG_NOBIP



Veja na figura acima, **(Run>Dictionaries)** as categorias criada para AlarmGroups e também para Nodes (em Devices) e para as tabelas Historian.

Para que sejam criados os registros de Alarm items para cada tag, bastará que no registro do tag, no caso na sua definição em **templates** seja aplicada esta **categoria**.

A aplicação de uma categoria à um objeto é utilizada de forma hereditária: isto é, se for aplicada à um objeto hierarquicamente superior, por exemplo no template de um TRAFO, todos os tags definidos neste template também terão a mesma categoria. São consideradas também de forma acumulativa as categorias atribuídas diretamente nos tags (folhas). Em outras palavras o tag (objeto de mais baixo nível, ou folha) tem a si atribuídas todas as categorias (um OU delas) atribuídas aos objetos de nível superior.

Points em Devices

O usuário deverá criar manualmente, os **channels** e **nodes** da Aplicação, de acordo com as características próprias dos protocolos utilizados na aplicação. Esta parametrização deverá ser feita em função dos IEDs com os quais o projeto vai se comunicar.

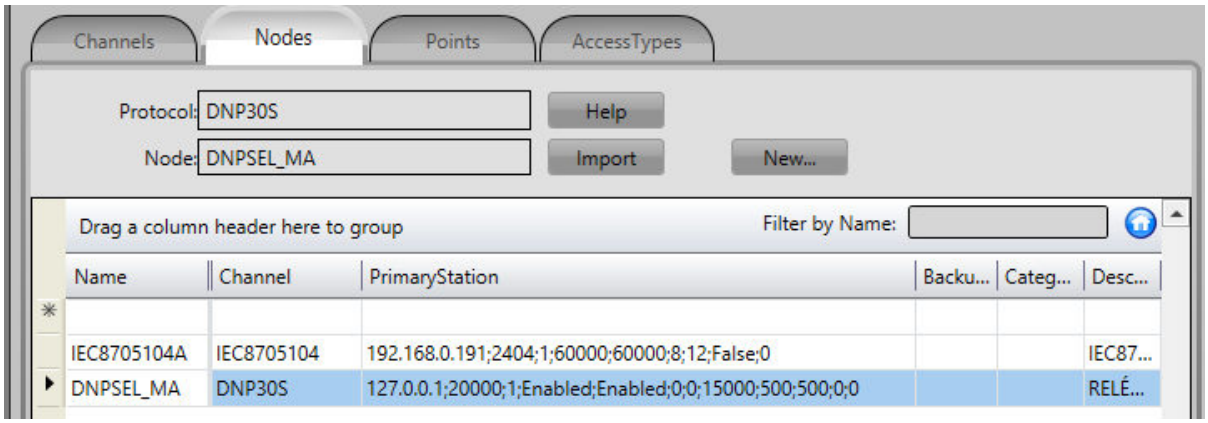
Para cada node existente, o sistema criará, automaticamente, uma categoria com o prefixo NODE_ seguido do nome do node. Por exemplo, para os nodes definidos na figuras abaixo **(Edit>Devices)** serão criadas as categorias:

NODE_IEC8705104A e

NODE_DNPSEL_MA

Estas categorias deverão ser atribuídas a todos os tags de pontos que são tratados por estes Nós de comunicação.

Tipicamente, quando se tem templates que definem objetos de nível mais alto, como um Alimentador, ou um Trafo, em que todos os pontos monitorados e controlados vem através de um único nó de comunicação., a atribuição da categoria de Nó a estes objetos facilita muito a tarefa.



Histórico

O usuário deverá criar todas as **historian tables** necessárias à aplicação. Por exemplo poderá criar três **historian tables** H1, H2 e H3 da seguinte forma:

- H1 - Salva registros a cada minuto;
- H2 - Salva registros a cada cinco minutos;
- H3 - Salva registros sempre que a variável tem seu valor alterado;

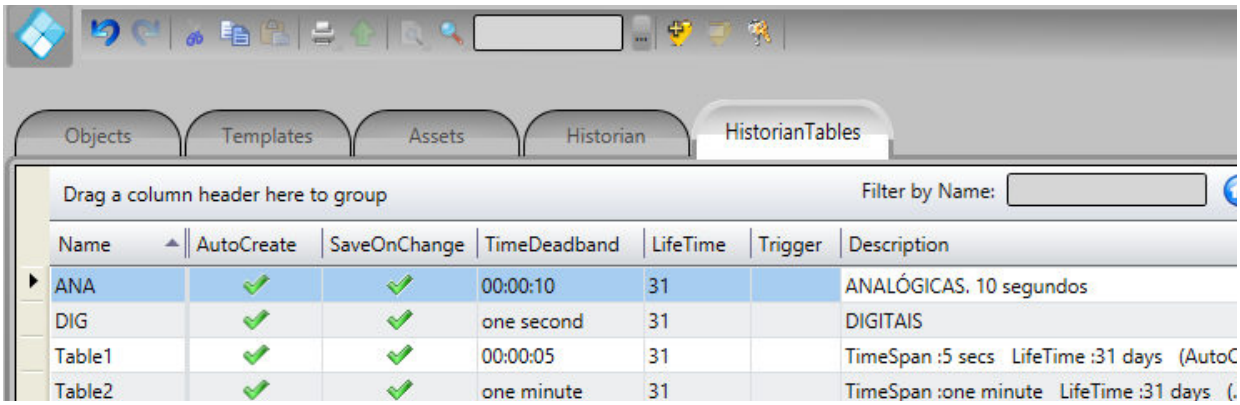
Para cada **historian table** criada, deverá ser criada uma categoria correspondente com o prefixo HIST_ seguido do nome da **historian table**.

Poderão ser criadas quantas **historian tables** forem necessárias, sem restrições de nomenclatura, e com qualquer parametrização de salvamento, porém deverá, obrigatoriamente, existir uma categoria HIST_ + nome da **historian table.**

Na figura abaixo (**EDIT > TAGS**) foram criadas as tabelas ANA e DIG. As categorias a serem criadas pelo usuário deverão ser: HIST_ANA e HIST_DIG.

Para cada um dos tags que se deseja tenham seus valores e estados gravados em cada uma destas tabelas de históricos, deverão ser aplicadas as **categorias** correspondentes**.**

Novamente aqui, lembre-se que o trabalho será simplificado se for possível aplicar as categorias à objetos de nível mais alto. Tal facilidade será decorrente diretamente da maneira que se faz a estruturação dos templates na modelagem dos objetos reais.



Atribuição de Categorias

No projeto padrão do cliente, conforme apresentado acima, estarão pré-criados: Alarm Groups, Templates, Channel, Nodes, Historian Tables, Dictionaries (Localization, Enumeration e Categories) e a Symbol Library associada a cultura do usuário.

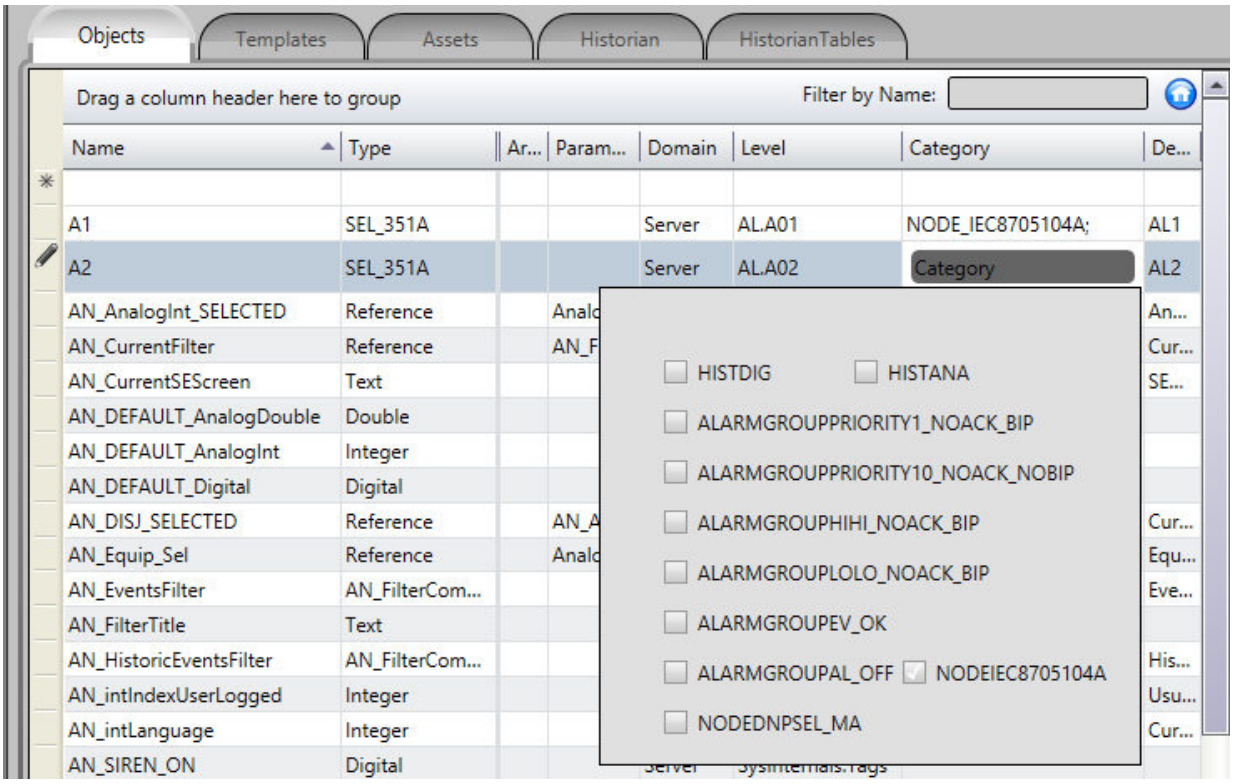
Para que sejam criados os registros nas tabelas, será necessário que sejam atribuídas aos tags nas tabelas **objects e templates** as categorias para cada um destas tabelas.

Feitas as atribuições de categorias, fica garantida a criação dos registros nas tabelas, porém os registros serão criados com alguns campos importantes deixados em branco para serem preenchidos posteriormente.

Uma facilidade adicional disponível no aplicativo **Deploy**, e´ a possibilidade de se especificar previamente os conteúdos para os atributos mais utilizados e importantes destes registros. Isto é feito pelo uso da coluna **comments** , disponível nas tabelas de **objects e templates.** Nestas colunas, para cada tag, pode-se especificar por meio de um texto com palavras chaves os conteúdos a serem utilizados na geração dos registros. Veja em "Atributos adicionais" na página 290, detalhes sobre este procedimento.

As categorias pré-definidas desejadas deverão ser atribuídas aos tags (folhas) nos diversos templates, conforme a figura abaixo. Observe-se que as categorias impostas para objetos de nível mais alto, definidos com Types do usuário (Templates) são herdadas pelos níveis inferiores nos objetos instanciados.

Veja na figura abaixo como impor uma categoria à um objeto. Clica-se na coluna **categorias** em **(Edit>Tags>Objetos),** e aparece a lista das categorias existentes. Pode-se escolher uma ou várias para um mesmo objeto, clicando nas caixinhas à esquerda de cada categoria. Na coluna **categorias**, fica o texto com o conjunto de nomes de **categorias** escolhidas, separadas por ponto e virgula (;).



Veja os exemplos a seguir:

1. Na figura acima escolheu-se para o objeto A1 (alimentador do tipo SEL_351A) a categoria NODE_IEC8705104A que define o node de comunicação com o relé. Neste template (type) estão definidas todas as tags de medição e status que se deseja ler neste IED.
2. Se a folha (tag) AMP de um template BAY, precisar criar um AlarmItem do tipo LO que exija reconhecimento e emita sinal sonoro, esta tag deverá ter associada a categoria ALARMGROUP_LO_ACK_BIP;
3. Se para a mesma tag se desejar que seus valores sejam salvos no histórico na tabela H2, deverá também possuir a categoria HIST_H2;
4. Se esta tag tiver endereço no campo e pertencer a um node de nome DNP1, deverá também possuir a categoria NODE_DNP1
5. Cada uma das categorias de diferentes tipos atribuídas serão utilizadas pelo aplicativo de Geração Automática de base, para a criação de registros correspondentes nas tabelas respectivas.

NOTA - Para que os tags sejam criados automaticamente, devem necessariamente ter atribuição de um **Level (assets),** e possuírem o atributo **Domain** como **server**. Estes atributos podem estar diretamente no tag ou em qualquer objeto de nível hierárquico superior, de forma que as folhas herdem estes atributos.

Criados todos os templates, feitas as categorizações das tags de cada template, preenchidos os campos comments desejados, o próximo passo será criar um objeto para instanciar o template desejado.

Exemplo: Para criar um trafo 1 a partir de um template de nome TRAFO:

1. Vá para **Edit>Tags>Objects** e crie um objeto

   Name=T1

   Type=TRAFO

   Level também deve ser obrigatoriamente preenchido.

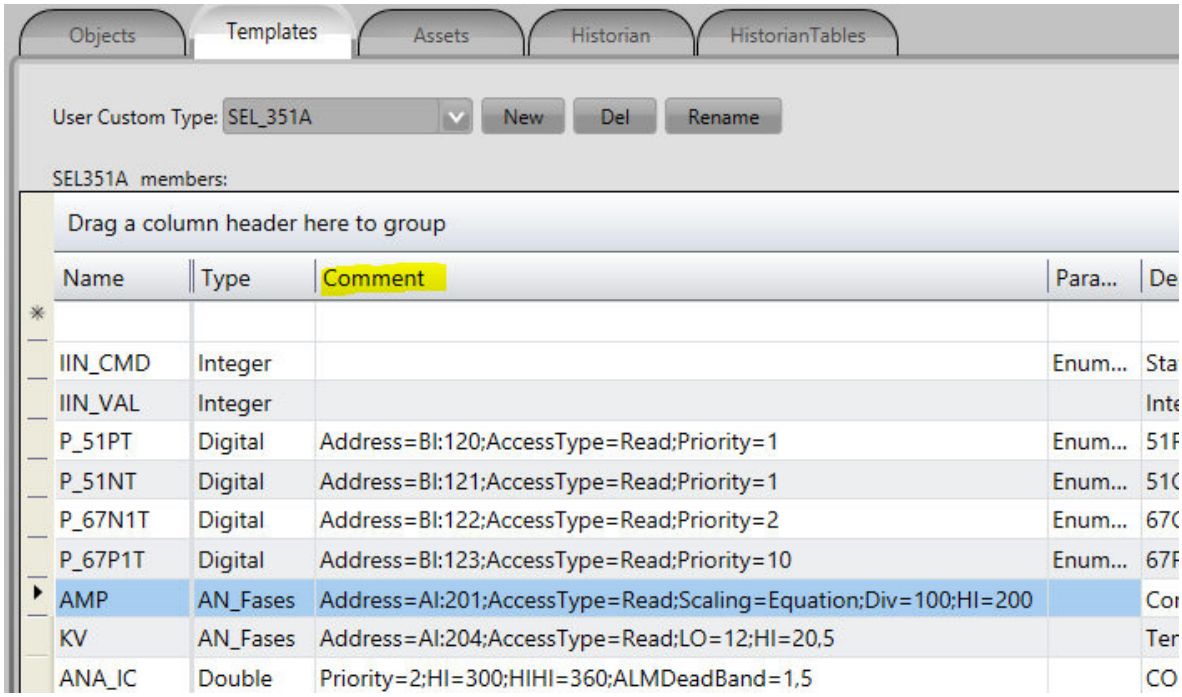
2. Caso for considerado adequado, pode-se ainda atribui outras **categorias** e definir **comments** à estes objetos de nível mais alto. Nestes níveis mais altos é conveniente a atribuição de categorias do tipo de NÓS de comunicação.

Atributos adicionais

Para a especificação de conteúdos previamente definidos, a serem usados nos atributos importantes existentes nos registros a serem criados nas tabelas pelo **Deploy,** está disponível a facilidade do uso dos campos **comment,** nas tabelas **objects e templates.**

Para cada atributo considerado, em cada uma das tabelas, foram criadas palavras chaves, que em sua maioria são os próprios nomes das colunas. Em cada tag, ou objeto de nível superior, pode-se digitar na coluna **comments,** textos formados com estas palavras chaves e os conteúdos que se deseja. O aplicativo **Deploy** utilizará estes conteúdos na geração dos registros relacionados nas tabelas **AlarmItems, Points e Historian.**

A figura abaixo mostra o uso da coluna **comments** em um template. Os items a seguir apresentam os atributos (colunas) considerados, as palavras chaves para cada tabela e breve descrição do uso.



Alarm Items

A tabela a seguir mostra as palavras chaves de colunas utilizadas.

Palavra chave	Coluna	Conteúdo
LO	LIMIT	Numero real, a ser utilizados em registros cujo grupo de alarme se refira a condição Lo (inferior operacional).
LOLO	LIMIT	Numero real, , a ser utilizados em registros cujo grupo de alarme se refira a condição LoLo (inferior emergencial)
HI	LIMIT	Numero real, , a ser utilizados em registros cujo grupo de alarme se refira a condição Hi (superior operacional)
HIHI	LIMIT	Numero real, , a ser utilizados em registros cujo grupo de alarme se refira a condição HiHi (superior emergencial)
EQUAL	LIMIT	Numero inteiro representativo do estado no qual o alarme fica ativo
NOTEQUAL	LIMIT	Numero inteiro representativo do estado no qual o alarme NÃO fica ativo
PRIORITY	PRIORITY	PRIORITY - Um numero inteiro de 1 a 10 a ser utilizado como prioridade de alarme para o item
ALM_DEADBAND	DeadBand	DeadBand - Um numero real para ser usado como banda morta na geração e normalização de alarme

Exemplo:

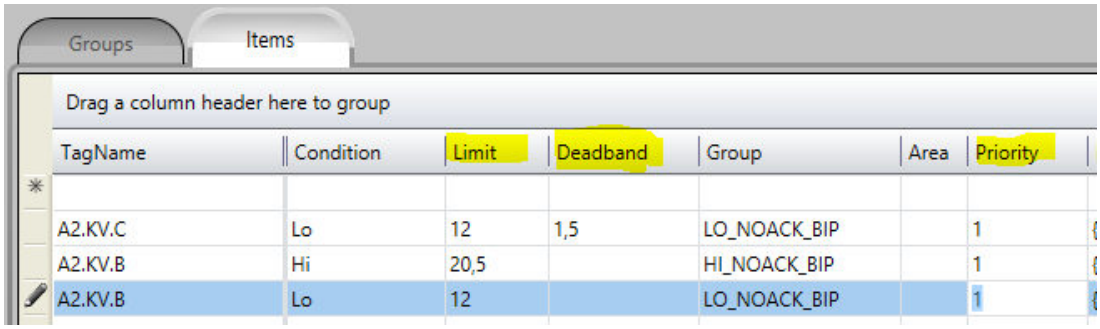
1. No campo **comment** do tag A2.KV.C teria sido colocado

   ```
   Priority=1;LO=12;Alm_DeadBand=1,5
   ```

2. No campo **comment** do tag A2.KV.B. teria sido colocado

   ```
   Priority=1;HI=20,5;LO=12
   ```

3. Veja na figura abaixo as colunas preenchidas com estas palavras chaves:



NOTA - Quando um tag ou objeto é definido por vários níveis de templates, o **comment** que será processado pelo **Deploy,** será formado pela concatenação de todos os **comments** especificados na hierarquia direta. No caso de mais de uma especificação com a mesma palavra chave, a que de fato será utilizada é a que for definida no nível mais baixo do trajeto.

Points

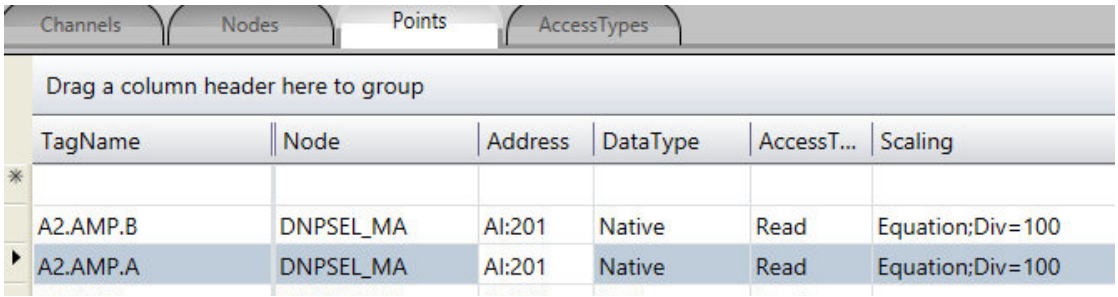
A tabela a seguir mostra as palavras chaves de colunas utilizadas.

Palavra chave	Coluna	Conteúdo
ACCESSTYPE	AccesType	Identificação do AccessType a ser utilizado. O conteúdo é sensível à caixa: usar maiúsculas e minusculas corretamente.
ADDRESS	Address	Texto com o conteúdo correto para o protocolo utilizado. Exemplo para um ponto digital em DNP30 com endereço 328 o texto deve ser: `BI:328`
SCALING	Scaling	O tipo de Scaling utilizado: `Linear ou Equation.`
Div	Scaling	Para o caso de scaling Equation, define o valor do divisor (real) e este texto deve seguir o tipo de Scaling
Add	Scaling	Para o caso do Scaling Equation, define o valor a ser somado (real)
EngValue1	Scaling	Para o caso do Scaling Linear. Valor real deve seguir o tipo de Scaling
EngValue2	Scaling	Para o caso do Scaling Linear. Valor real deve seguir o tipo de Scaling
DeviceValue1	Scaling	Para o caso do Scaling Linear. Valor real deve seguir o tipo de Scaling
DeviceValue2	Scaling	Para o caso do Scaling Linear. Valor real deve seguir o tipo de Scaling

Exemplo:

1. No campo **comment** do objeto A2.AMP (que tem um subtype Fases) teria sido colocado

```
ADDRESS=A1:201;ACCESSTYPE=Read;SCALING=Equation;Div=100
```



Historian tables

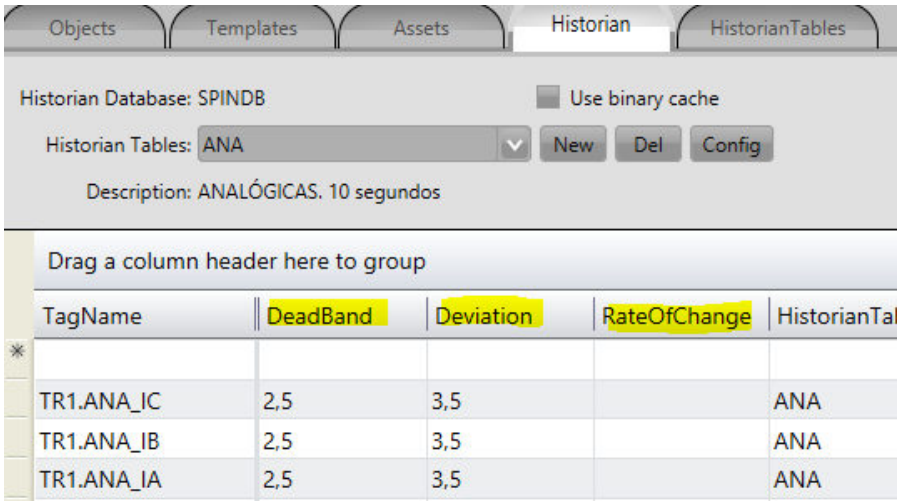
A tabela a seguir mostra as palavras chaves de colunas utilizadas.

Palavra chave	Coluna	Conteúdo
DEVIATION	Deviation	Valor real de desvio em relação ao anterior para nova gravação
HIST_DEADBAND	DeadBand	Valor real, Banda morta em relação ao valor anterior
RATEOFCHANGE	RateOfChange	Taxa de variação máxima em relação ao valor anterior, para nova gravação. real.

Exemplo

1. No campo **comment** do objeto TR1 teria sido colocado

```
HIST_DEADBAND=2,5;DEVIATION=3,5
```

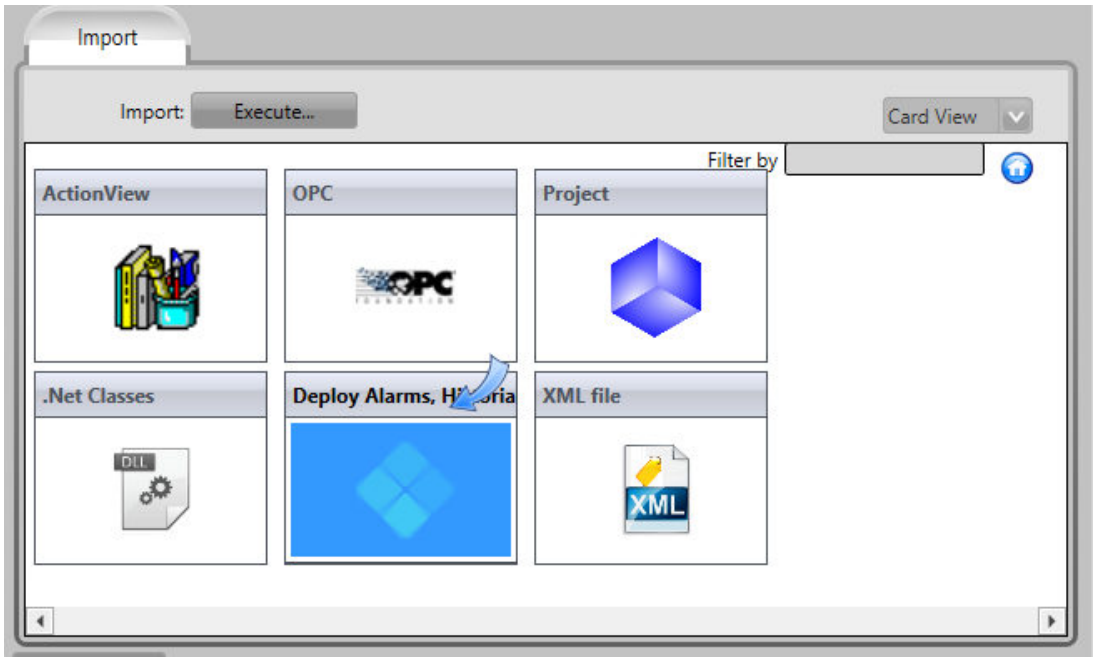


Executando o Deploy

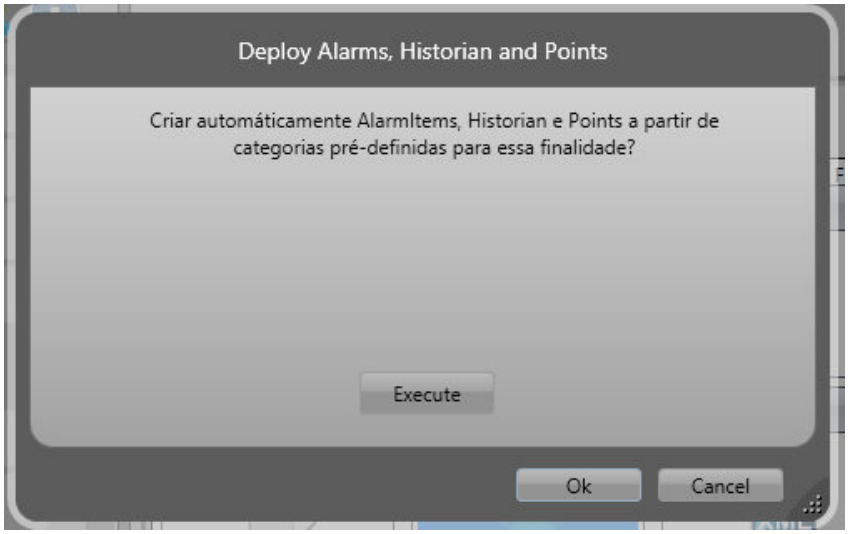
Para fazer a geração automática do projeto (explodir), deverá ser executada a extensão Deploy que procurará todos objects que não foram criados e os criará.

Vá para Run>Extensions

1. Escolha o simbolo **Deploy;**



2. Ao chamar esta extensão aparece uma janela para o acompanhamento da execução.



3. Clique no botão *Execute*, e acompanhe a criação de cada uma das tabelas e seus registros sendo executada.

A verificação se o objeto foi ou não já criado é feita automaticamente pelo Deploy que verifica se já existe um elemento com mesmo nome e, se positivo, não o recria pela segunda vez.

Serão criados os registros para as seguintes tabelas:

Tags/Historian

Device/ Points

Alarm/Items

Observações:

1. Em devices / points são criadas todas as entradas e saídas associadas aos nodes. É preciso completar a tabela com os endereços e os AccessTypes. Exportar do ActionNET para uma planilha, copiar os endereços de uma planilha Excel de endereços feita pelo cliente e importa-la novamente para o ActionNET.
2. Na tabela Alarms e na definição dos templates os nomes e siglas são importantíssimos pois eles são parte da cultura do cliente. Os Limites para alarmes HI, LO, HIHI e LOLO devem ser colocados nos itens manualmente.
3. Se por algum motivo se desejar recriar a tabela Historian, deve-se remover todos os itens já criados e, no diretório do projeto, remover o arquivo com o nome do mesmo projeto e sufixo *_DeployHistorian_HandledTags.xml*.

.

Glossário

Cada área de conhecimento tem sua própria terminologia, o mesmo se aplica a aplicativos de software. Certamente, a nomenclatura Action.NET usa segue padrões de mercado, sempre que possível, mas existem termos específicos relacionados ao sistema e outras palavras podem ter um significado especial quando se utiliza neste contexto.

Este glossário tem a maioria dos termos e palavras-chave usadas pelo Action.NET e também conceitos e tecnologias relacionadas que são relevantes para o nosso contexto.

É extremamente recomendado pelo menos ler deste glossário uma vez, depois que voce tenha usado o Action.NET por um ou dois meses. Fazendo uma revisão das definições e nomenclatura utilizadas pela plataforma Action.NET e as equipes de engenharia e desenvolvimento da SPIN, combinados com a sua experiência com o produto, fará com que seu entendimento do sistema o capacite para a criação de aplicativos de um mais alto nível.

App Domain or Application Domain	Representa o ambiente do computador isolado virtual, onde uma aplicação dot NET é executada. Isolado, neste contexto, significa que as áreas de memória e componentes utilizados pela programação em execução nesse ambiente são completamente isolados de outros programas em execução no mesmo computador. O isolamento inerente dos domínios dot NET cria uma proteção intrinsecamente segura para os projetos em execução, uma vez que impede um processo de interferir com o outro, permitindo até mesmo que duas versões diferentes do mesmo programa possam ser executadas lado a lado. Uma definição mais técnica, de AppDomain é o espaço que é formado pelos limites que o CLR (Common Language Runtime) fornece em torno dos objetos criados a partir do mesmo aplicativo.
Application ou Aplicação	Refere-se a solução ou sistema final criado usando Action.NET. Os termos Aplicação e Projeto são usados o m significados próximos: Projeto refere-se a configuração de engenharia e arquivos necessários para implementar a solução e aplicação refere-se a solução conceitual que foi criada.
Assembly ou Montagem	Refere-se a um arquivo executável (. EXE ou Library.dll) criado usando código gerenciado e Microsoft NET framework..
Designer	Componente do Workspace de Engenharia usado para desenhar e animar monitores e telas de sinópticos.
DLL Hell	Termo usado para se referir as complicações na execução das aplicações legadas do Windows, pois não há isolamento embutido nas bibliotecas DLLs, memória e objetos COM usados pelas aplicações. O dot NET Framework elimina completamente este problema.

Domain, Server Domain, Client Domain ou Domínio, Domínio Servidor , Domínio Cliente	Refere-se aos valores e localização de objetos em tempo de execução. Objetos do domínio servidor são os objetos em execução no servidor durante a execução. Os valores associados a este objeto são de todo o sistema. Domínio do Cliente são os objetos específicos que estão sendo executado na estação cliente e cada máquina pode ter valores diferentes para eles.
Internal Module ou Módulo interno	São os recursos e os programas que implementam as funções internas do sistema, que são executadas sem qualquer configuração do usuário. Exemplos disso são a tarefa de Sincronização de Rede e o Gerador de Relatórios (executado em background). Algumas ferramentas de Estúdio, por exemplo, o ModuleInformation.exe, pode exibir informações sobre o status desses módulos internos, no entanto, esta informação é necessária apenas para otimizações avançadas do sistema.
Main Project File ou Arquivo principal de Projeto	Refere-se ao banco de dados SQL codificado que contém a configuração do projeto. O extension.TPROJ refere-se ao projeto atual ainda em desenvolvimento. As extension.TRUN refere-se a projetos publicados, em arquivos somente leitura.
Modifier (Device/ Points)	Parâmetro auxiliar para especificar modificações em pontos de leitura e escrita para um dispositivo, como o tratamento de Arrays, máscaras de bits, strings, inversão de bytes e outras operações em que a definição DataType não pode defini-las completamente.
Module	Um programa que acessa à base de dados em tempo real (BDTR) e pode ser composta de uma ou mais montagens.
Namespace	Um espaço de endereçamento. Todos os objetos criados com um namespace tem nomes exclusivos, unívocos. Espaço de endereço também pode ser entendido como uma forma de amarrar um conjunto de objetos. Todas as variáveis do processo são agrupadas no namespace denominado Tag; todos os relatórios estão agrupados no namespace Report. O nome de um objeto deve ser único dentro do namespace ao qual o objeto pertence.
Object Type (RunObj type)	Determina o tipo de objeto em tempo de execução (visualização, relatório script).
Objects, Runtime Objects	Objetos em tempo de execução são aqueles objetos visíveis (através de seus nomes) para acessar o projeto através de scripts e displays, como Tags, Reports Alarms, Displays, entre outros. Os namespaces também identificam qual módulo de execução é responsável para atualizar esses valores, por exemplo, o módulo de alarme irá iniciar e irá atualizar os valores dos objetos de tempo de execução associados aos alarmes.
ObjectValues, PropertyValue	Objetos em tempo de execução (por exemplo, Tags, deviceNodes, etc) podem ter uma ou mais propriedades definidas. A TAG (e o valor da propriedade), por exemplo, MIN, MAX, Descrição, etc., entre outras propriedades.

PropertyWatch	Utilitário de diagnóstico para verificar e modificar os valores dos objetos em tempo de execução.
Parameters(Tag Configuration)	Definição dos parâmetros de comportamento e de processamento para valores de Tags
Project	Um conjunto de configurações de estúdio, esboços de telas, relatórios, scripts e notas de usuários criado e editado com uma única entidade.
projectDB / ProjectDatabase	ProjectDB (ou banco de dados do projeto) é um banco de dados que contém informações de configuração. Correspondem aos arquivos com a extensão <Project>. Tproj (tipo atual, por exemplo) ou <Project> _ version.Teng (tipo liberado para uso, por exemplo).
Property or Atribute	Propriedade (value) associado a um Tag ou objeto.
RunDB, Runtime Database/RtDB	O banco de dados em tempo real criado quando o projeto está sendo executado. Todos os objetos que podem ser acessados durante a executam (como tags, Displays, relatórios) são objetos no banco de dados Runtime, também chamado de Banco de Dados de Tempo Real ou BDTR.
Runtime	Refere-se a um projeto com módulos carregados e em plena execução.
Runtime Startup	Operação que é faz o projeto entrar em tempo de execução. Esta operação pode ser realizada a partir do programa TStartup.exe na versão de atualização do Action.NET ou TServer.exe na versão publicada do projeto.
Tag	A variável de processo. Nome de um Namespace, que inclui todas as variáveis criadas pelo usuário em uma configuração de projeto.
Tag Type	Define o tipo de objetos no namespace Tag: Digital, analógico, texto. Estas tags são uma classe de componentes ou propriedades acessados diretamente, como mínimo, máximo, valor, qualidade. Cada propriedade é criada internamente como ValueType.
Task (Script.Task)	Programa escrito em VB.NET (ou C #), que é executado no servidor ou cliente durante o tempo de execução de um projeto. A execução será no servidor ou cliente, dependendo da configuração da propriedade de domínio no script.
TManager	O programa que deve ser executado para realizar a configuração de um projeto.
Toggle	Inverter o valor de uma variável. Valores maiores do que zero são convertidos para zero; zero é convertido para o valor "1".

Trust, Partial Trust, Full Trust ou Confiança, confiança parcial, Confiança completa	Confiança Parcial: Meio ambiente em que um aplicativo tem acesso limitado a recursos (acesso restrito aos arquivos de pastas particulares, executando outras aplicações, etc.) Aplicativos XBAP que rodam dentro de um navegador (por exemplo, Internet Explorer) devem ser executados em "Confiança Parcial". Confiança Total: Meio ambiente em que um aplicativo tem acesso a todos os recursos do sistema. Aplicativos instalados em um computador geralmente são executados em modo "Full Trust".
Visibility (Tag Visibility)	Refere-se ao sistema de visibilidade de Tags; Tags podem ser privados, públicos ou protegidos. Público: O valor da Tag durante a execução do Runtime está disponível para o acesso a programas externos através de TCP / IP ou OPC Server. Além disso, o valor do Tag é necessariamente global, ou compartilhado entre todas as estações clientes (Domínio do servidor). Protegido: Somente leitura. Particular: A Tag definido como "particular" não pode ser acessado por aplicações externas (OPC Server, TCP / IP) e será executado em máquinas cliente ou servidor de acordo com a configuração do aplicativo com as seguintes características: • Tags usadas apenas em módulos chamados pelo cliente (como Telas e Reports) executados no escopo do cliente, ou podem ter valores diferentes em cada máquina do cliente (Cliente de Domínio); • Tags usadas em módulos do servidor, tais como dispositivos e alarmes têm um valor único no sistema (Servidor de Domínio).
XBAP	**Xaml Browser Application (XBAP)** é um aplicativo gráfico usando a tecnologia XAML que é executado dentro de um navegador (Internet Explorer).